

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STROJNÍCKA FAKULTA**

**MAGNETOSHIELD: EXPERIMENTÁLNY PRÍSTROJ NA MAGNETICKÚ
LEVITÁCIU OBJEKTOV**

Diplomová práca

SjF-5226-81742

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
STROJNÍCKA FAKULTA**

**MAGNETOSHIELD: EXPERIMENTÁLNY PRÍSTROJ NA MAGNETICKÚ
LEVITÁCIU OBJEKTOV**

Diplomová práca

SjF-5226-81742

Študijný odbor: Kybernetika
Študijný program: 2621T16 I-AISP automatizácia a informatizácia strojov a procesov
Školiace pracovisko: Ústav automatizácie, merania a aplikovanej informatiky
Vedúci záverečnej práce: prof. Ing. Gergely Takács, PhD.
Konzultant: Ing. Erik Mikuláš

Bratislava, 2021

Bc. Jakub Mihalík



ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Jakub Mihalík**
ID študenta: 81742
Študijný program: automatizácia a informatizácia strojov a procesov
Študijný odbor: kybernetika
Vedúci práce: prof. Ing. Gergely Takács, PhD.
Konzultant: Ing. Erik Mikuláš
Miesto vypracovania: ÚAMAI SjF STU v Bratislava

Názov práce: **MagnetoShield: Experimentálny prístroj na magnetickú levitáciu objektov**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Úlohou študenta je pokračovať v návrhu prototypu didaktického hardvéru a softvéru na výučbu konceptov automatického riadenia a mechatroniky. Jedná sa o miniaturizovaný modul na magnetickú levitáciu drobných objektov. Študent analyzuje druhú (poslednú) verziu experimentálneho zariadenia tzv. MagnetoShield R2. Na základe získaných informácií navrhne novú, tretiu verziu hardvéru MagnetoShield R3. Taktiež vytvorí prostredie aplikačného programovania (application programming interface, API) v jazykoch C/C++, prípadne MATLAB a Simulink spolu s príkladmi jeho využitia pri riadení a analýze fyzikálneho modelu zariadenia.

V rámci diplomovej práce ďalej skúma:

- existujúci fyzikálny model zariadenia a jeho presnosť, prípadný návrh nového presnejšieho modelu,
- možnosti riadenia založené na vytvorenom fyzikálnom modeli ako sú riadenie pomocou uloženia pólov alebo LQ regulátor,
- návrh lineárneho MPC regulátora s obmedzeniami na vstupných veličinách.

Rozsah práce: 50-70 s.

Riešenie zadania práce od: 15. 02. 2021

Dátum odovzdania práce: 28. 05. 2021

Bc. Jakub Mihalík
študent

prof. Ing. Cyril Belavý, CSc.
vedúci pracoviska

prof. Ing. Cyril Belavý, CSc.
garant študijného programu

Čestné prehlásenie

Vyhlasujem, že som záverečnú prácu vypracoval samostatne s použitím uvedenej literatúry.

Bratislava, 28. mája 2021

.....
Vlastnoručný podpis

V prvom rade sa chcem podakovať vedúcemu práce prof. Ing. Gergelymu Takácsovi za umožnenie práce na projekte MagnetoShield, ktorý je súčasťou iniciatívy AutomationShield, ako aj za usmerňovanie a technickú výpomoc.

Ďalej by som sa rád podakoval kolegom študentom, ktorí boli súčasťou tejto iniciatívy. Ich práca bola pre mňa mnoho krát veľkou pomocou, ktorou som sa mohol inšpirovať alebo z nej čerpať.

Na záver by som chcel podakovať všetkým vzdelávacím inštitúciám, ktoré umožňujú bezplatný a jednoduchý prístup ku kvalitným zdrojom informácií v podobe zaznamenaných prednášok, náučných videí alebo rôznych rezumé z odborných tém. Veľká vďaka za to, že poskytujú tieto informácie nie len študentom, ale aj intelektuálnej verejnosti, túžiacej po poznaní. Ich vízia bezplatného osobného vzdelávania a slobodného prístupu k pravdivým a overeným informáciám je inšpiratívna a príkladná pre všetky vzdelávacie inštitúcie.

Názov práce: MagnetoShield: Experimentálny prístroj na magnetickú levitáciu objektov

Kľúčové slová: elektromagnetická levitácia, AutomationShield, MagnetoShield, Arduino, Simulink, modelovanie, identifikácia, linearizácia, spätnoväzobné riadenie, stavový priestor, kvadratické programovanie, LQR, prediktívne riadenie

Abstrakt: Cieľom tejto práce je čitateľovi predstaviť zariadenie na magnetickú levitáciu MagnetoShield, spolu s možnosťami jeho využitia za účelom vzdelávania v oblasti teórie riadenia. Na úvod je čitateľ oboznámený s tým, o aké zariadenie sa jedná, na akom princípe funguje a taktiež je predstavená stručná história vývoja tejto edukačnej pomôcky. Ďalej je detailnejšie opísaný hardvér najnovšieho modelu tohto zariadenia, MagnetoShield R4. S hardvérom sa spája aj téma, akým spôsobom je používateľ schopný opísané hardvérové prvky ovládať. Za týmto účelom práca predstavuje detailný opis softvérových rozhranií pre interakciu používateľa s MagnetoShieldom. Ide o Arduino IDE, pre ktoré bola napísaná knižnica v jazyku C/C++ a Simulink, pre ktorý bola taktiež vytvorená samostatná knižnica. Okrem opisu funkcií, ktoré sú prístupné používateľovi, práca uvádza aj postup, ako je možné vytvorené programy v Simulinku nahrať na prototypizačný hardvér - v tejto práci ním je Arduino Due. Ďalej text pojednáva o problematike modelovania systému elektromagnetického levitátora. Následne je na základe matematicko-fyzikálnych úvah navrhnutý model takéhoto systému v podobe dvoch nelineárnych diferenciálnych rovníc. Nakoľko cieľom práce je aj deklarovať použiteľnosť navrhnutého modelu za pomocí lineárneho spätnoväzobného riadenia, práca detailne popisuje proces linearizácie. Keďže model nemusí presne reprezentovať skutočnosť, ďalej je v práci opísané, ako zvýšiť percentuálnu zhodu navrhnutého modelu s reálnym správaním sa systému pomocou procesu identifikácie. V predposlednej časti práce sú opísané teoretické základy pre návrh spätnoväzobných riadiacich algoritmov sledujúcich referenciu. Taktiež je uvedený a popísaný koncept stavového pozorovateľa a Kalmanovho filtra. Následne sú uvedené jednotlivé algoritmy so svojimi charakteristickými črtami a spôsob ich implementácie pomocou softvérového rozhrania Simulink za použitia uvedenej teórie. Menovito sa jedná o riadenie PID regulátorom, uložením pólov, LQ regulátor a prediktívne riadenie. V závere práce je uvedených ešte niekoľko dodatkov, venujúcich sa konceptu kvality navrhnutého modelu a jeho nelineárnej forme. V úplnom závere je možné nájsť zhodnotenie práce a návrhy v prípade ďalšieho vývoja zariadenia MagnetoShield.

Title: MagnetoShield: An Experimental Device for Magnetic Levitation

Key words: electromagnetic levitation, AutomationShield, MagnetoShield, Arduino, Simulink, system modelling, identification, linearization, feedback control, state space representation, quadratic programming, LQR, predictive control

Abstract: This thesis presents an electromagnetic levitation device, referred to as the MagnetoShield, and demonstrates several usage possibilities for the education of control theory. The main principles of the device and the short history of its development are presented in the beginning. The work continues with the hardware description of the newest model, MagnetoShield R4, and discusses various software interfaces allowing the user to interact with the device. For this purpose, a C/C++ library has been created, which is compatible with the Arduino IDE, along with a library for Simulink. The work contains a detailed description of the functions of both libraries. The next major part of the thesis focuses on the problem of system modelling. Mathematical and physical analysis is used to find and design a well-fitting model. The proposed model is in the form of two nonlinear differential equations. Consequently, the thesis describes a linearization process and a parameter identification procedure. The reader is presented the fundamentals of feedback control with reference tracking, along with the concepts of state observer and the Kalman filter. The aforementioned theoretical concepts are applied to the design of four control algorithms – PID, pole placement, LQR and predictive control. Several additional topics focusing on the quality of the designed model are mentioned in the final chapter, including testing its natively nonlinear form. The thesis is concluded by providing guidelines for further research and development of the MagnetoShield.

Obsah

Zoznam obrázkov	4
Zoznam premenných a veličín	7
1 Úvod	10
2 Vývoj zariadenia MagnetoShield	12
2.1 MagnetoShield R1	13
2.2 MagnetoShield R2	15
2.3 MagnetoShield R3	19
3 MagnetoShield R4 – Hardvér	20
4 MagnetoShield R4 – Softvérové rozhrania	27
4.1 Arduino IDE	27
4.2 MATLAB a Simulink	32
4.2.1 Konfigurácia Simulinku pre Arduino DUE	39
4.3 Python	42
5 Modelovanie a identifikácia	43
5.1 Základný koncept	44
5.2 Silové zložky pohybovej rovnice	45
5.3 Elektrická rovnica	49
5.4 Lineárny model	52
5.5 Identifikovanie lineárneho modelu	56
6 Riadenie	66
6.1 Stavové spätnovázobné riadenie	67
6.2 Sledovanie referencie	70
6.3 Odhad stavov	72
6.4 PID	76
6.5 Riadenie uložením pólov	79
6.6 LQ regulátor	89
6.7 Prediktívne riadenie	98
7 Dodatočné príklady	108
7.1 Modely systému MagnetoShield	108
7.2 Nelineárny model	115
8 Záver	120

Zoznam obrázkov

2.1	MagnetoShield R1	14
2.2	Elektrický obvod zariadenia MagnetoShield R1	14
2.3	MagnetoShield R2	16
2.4	Elektricky obvod zariadenia MagnetoShield R2	17
2.5	Závislosť napäťia od DAC levelov na elektromagnete.	18
3.1	MagnetoShield R4	20
3.2	Schéma zosilňovača na MagnetoShield R4	21
3.3	Schéma merania napäťia na elektromagnete pre MagnetoShield R4	22
3.4	Závislosť napäťia na elektromagnete od DAC levelov.	23
3.5	Schéma elektrického obvodu MagnetoShield R4	25
4.1	Blok <i>magnetoShieldPins</i>	34
4.2	Dialógové okno po otvorení bloku <i>magnetoShieldPins</i>	34
4.3	Blok <i>actuatorWrite</i>	35
4.4	Dialógové okno po otvorení bloku <i>actuatorWrite</i>	35
4.5	Blok <i>outputConverter</i>	36
4.6	Dialógové okno po otvorení bloku <i>outputConverter</i>	36
4.7	Blok <i>inputConverter</i>	37
4.8	Dialógové okno po otvorení bloku <i>inputConverter</i>	37
4.9	Blok <i>sensorRead</i>	38
4.10	Dialógové okno po otvorení bloku <i>sensorRead</i>	38
4.11	Blok <i>MagnetoShield</i>	39
4.12	Dialógové okno po otvorení bloku <i>MagnetoShield</i>	39
4.13	Postup konfigurácie dosky Arduino pri používaní MagnetoShieldu v prostredí Simulink.	41
5.1	Sily pôsobiace na levitujúci magnet.	45
5.2	Základné rozmery pri výpočte počtu závitov cievky.	47
5.3	Súradnicový systém a vektorové veličiny.	49
5.4	Súradnicový systém a orientácia sín.	49
5.5	Zjednodušenie elektrického obvodu pre napájanie elektromagnetu.	50
5.6	Priebeh meraných stavov a vstupu.	58
5.7	Priebeh stavov a vstupu po odrátaní linearizačných bodov.	59
5.8	Vstupná veličina a stavové veličiny reprezentované vo frekvenčnej oblasti.	60
5.9	Porovnanie modelu systému s nameranými dátami pred odhadom parametrov.	62
5.10	Porovnanie identifikovaného modelu s nameranými dátami.	64
6.1	Grafické znázornenie diferenciálnej rovnice.	67
6.2	Grafické znázornenie diferenčnej rovnice.	69

6.3	Grafické zobrazenie rovnice dynamiky stavovej reprezentácie.	69
6.4	Schéma zápornej spätej väzby stavového modelu.	70
6.5	Sledovanie referencie pomocou násobiteľa N	71
6.6	Sledovanie referencie pomocou integračného stavu $x_i(t)$	72
6.7	Koncept pozorovateľa a získania odhadu stavov \hat{x}	73
6.8	Schéma Luenbergerovho pozorovateľa.	74
6.9	Schéma riadenia PID regulátorm.	78
6.10	Priebeh regulovanej veličiny pri riadení PID regulátorom.	79
6.11	Priebeh stavov s umiestnenými pólmi α_1	82
6.12	Priebeh stavov s umiestnenými pólmi α_2	83
6.13	Vplyv voľby pólov na dynamiku pozorovateľa.	85
6.14	Schéma simulačného programu pre komplexné testovanie funkčnosti navrhnutého riadenia uložením pólov.	86
6.15	Simulácia riadenia umiestnením pólov s použitím Luenbergerovho pozorovateľa uvažujúca prítomnosť šumu. "Modelové dátá" reprezentujú simulovaný signál z modelu so zakomponovaným šumom, "Odhad" je výstup z pozorovateľa.	87
6.16	Priebeh riadenia umiestnením pólov s použitím Luenbergerovho pozorovateľa.	88
6.17	Schéma riadenia uložením pólov.	88
6.18	Simulácia riadenia LQ regulátorm s použitím Kalmanovho filtra uvažujúca prítomnosť šumu. "Modelové dátá" reprezentujú simulovaný signál z modelu so zakomponovaným šumom, "Odhad" je výstup z filtra.	96
6.19	Priebeh riadenia LQ regulátorm s meraním jediného stavu a s použitím Kalmanovho filtra.	97
6.20	Schéma riadenia LQ regulátorm.	97
6.21	Schéma riadenia polohy levitácie na základe prediktívneho riadenia.	102
6.22	Priebeh stavových veličín a vstupu pri MPC riadení s predikčným horizontom $N = 20$	103
6.23	Priebeh stavových veličín a vstupu pri MPC riadení s predikčným horizontom $N = 15$	104
6.24	Priebeh stavových veličín a vstupu pri MPC riadení s predikčným horizontom $N = 10$	105
6.25	Priebeh riadenia pomocou eMPC.	107
6.26	Schéma riadenia pomocou eMPC.	107
7.1	Porovnanie modelu z článku s dátami z PID regulácie bez vykonania odhadu parametrov.	109
7.2	Porovnanie navrhnutého modelu z Kap. 5 s dátami z PID regulácie bez vykonania odhadu parametrov.	110
7.3	Porovnanie modelu z článku s dátami z PID regulácie po vykonaní odhadu parametrov.	111
7.4	Porovnanie navrhnutého modelu z Kap. 5 s dátami z PID regulácie po vykonaní odhadu parametrov.	112
7.5	Simulované PID riadenie s použitím lineárneho modelu systému.	112
7.6	Priebeh simulácie PID regulácie modelu z článku.	113
7.7	Priebeh simulácie PID regulácie navrhnutého modelu z Kap. 5.	114
7.8	Nelineárny model systému v prostredí Simulink pre spojitý čas CT.	115
7.9	Nelineárny model systému v prostredí Simulink pre diskrétny čas DT.	116
7.10	PID riadenie s "nelineárnym pozorovateľom".	117

7.11	Priebehy nameraných stavov a ich odhadnuté hodnoty pomocou nelineárneho pozorovateľa.	118
7.12	Simulované PID riadenie nelineárneho modelu systému.	118
7.13	Priebeh simulácie PID riadenia nelineárneho modelu MagnetoShieldu s použitím ladenia zo skutočného systému.	119

Zoznam fyzikálnych veličín

Symbol	Veličina	Jednotka (iná jednotka)
A	systémová matica v CT stavovom priestore	-
A_d	systémová matica v DT stavovom priestore	-
B	matica vstupov v CT stavovom priestore	-
B_d	matica vstupov v DT stavovom priestore	-
\vec{B}	magnetická indukcia	T (G)
\vec{B}_r	remanencia	T
c	konštantá tlmenia	$\text{kg}\cdot\text{s}^{-1}$
C	matica výstupov v CT stavovom priestore	-
C_d	matica výstupov v DT stavovom priestore	-
D	matica priamej väzby vstupu na výstup v CT stavovom priestore	-
D_d	matica priamej väzby vstupu na výstup v DT stavovom priestore	-
\vec{F}	výsledná sila pôsobiaca na systém	N
\vec{F}_g	tiažová sila	N
\vec{F}_{mag}	sila medzi magnetom a elektromagnetom	N
\vec{F}_v	sila od tlmenia	N
$g(\vec{x})$	obmedzenie nerovnosti	-
\vec{g}	tiažové zrýchlenie	$\text{m}\cdot\text{s}^{-2}$
G	funkcionál prenosu	-
h	výška levitácie	m (mm)
$h(\vec{x})$	obmedzenie rovnosti	-
H	Hemiltonián, Hessián	-,-
I	elektrický prúd	A (mA)
J	účelová funkcia	-
\tilde{J}	rozšírená účelová funkcia	-
k	konštantá modelu	-
K	magnetická konštantá systému, zosilnenie stavov systému v spätej väzbe	$\text{N}\cdot\text{A}^{-1}$
K_D	derivačná zložka PID regulátora	-
K_e	magnetická konštantá systému pre elektrickú rovnicu	$\text{N}\cdot\text{A}^{-1}$
K_I	integračná zložka PID regulátora	-
K_{kal}	Kalmanovo zosilnenie	-
K_m	magnetická konštantá systému pre pohybovú rovnicu	$\text{N}\cdot\text{A}^{-1}$
K_P	proporčná zložka PID regulátora	-
l	dĺžka cievky elektromagnetu	m

L	indukčnosť cievky, zosilnenie pozorovateľa	H, -
$L_k(\vec{x}_k, \vec{u}_k)$	funkcia váhujúca členy počas doby po dosiahnutie horizontu	-
m	hmotnosť magnetu	kg
\vec{m}_i	magnetický moment dipólu	A·m ²
\vec{M}	magnetizácia	A·m ⁻¹
N	počet závitov, zosilnenie integračných stavov, dĺžka horizontu	-, -, -
p	konštanta modelu, póly systému	-, -
P_k	kovariancia chyby	-
Q	penalizačná matica stavov	-
r	referenčný signál	-
\vec{r}	poloha magnetu vzhľadom na elektromagnet	m (mm)
\vec{r}	smerový vektor osi	
R	elektrický odpor, polomer cievky elektromagnetu, penalizačná matica vstupov	Ω, m, -
s	citlivosť hallovho senzoru	V·G ⁻¹
\vec{S}	vektor plochy	m ²
S	veľkosť plochy, matica penalizujúca stav na horizonte	m ² , -
t	čas	s (ms)
T_d	derivačná časová konštantá	-
T_i	integračná časová konštantá	-
T_s	vzorkovacia períoda	s (ms)
u	vstupný signál – napätie	V
u_0	linearizačný bod pre vstupný signál – napätie	V
\vec{u}_k	vektor vstupných signálov v k -tom okamihu	V
U	elektrické napätie, funkcionál vstupu do systému	V, -
U_{DAC}	žiadane elektrické napätie na DAC prevodníku, vyjadrené ako bitové číslo	-
$\Delta u(t)$	odchýlka napäťia od u_0	V
V	objem	m ³
v	šum merania, rýchlosť	-, m·s ⁻¹
w	procesný šum	-
x	neznáma, stavová premenná	-
\vec{x}_k	vektor stavov pre k -ty okamih	-
x_1	stavová veličina poloha	m (mm)
x_2	stavová veličina rýchlosť	m·s ⁻¹
x_3	stavová veličina prúd	A (mA)
x_i	integračná stavová veličina	-
x_{01}	linearizačný bod pre polohu	m (mm)
x_{02}	linearizačný bod pre rýchlosť	m·s ⁻¹
x_{03}	linearizačný bod pre prúd	A (mA)
\hat{x}	odhad stavu	-
$\Delta x_1(t)$	odchýlka polohy od x_{01}	m (mm)
$\Delta x_2(t)$	odchýlka rýchlosťi od x_{02}	m·s ⁻¹
$\Delta x_3(t)$	odchýlka prúdu od x_{03}	A (mA)
X	funkcionál výstupu zo systému	-
y	meraný výstupný signál	-
z	regulačná odchýlka, chyba odhadu	-

α	žiadane póly systému	-
μ, μ_0	permeabilita materiálu, vákua	$\text{H}\cdot\text{m}^{-1}$
μ_r	relatívna permeabilita	-
λ	Lagrangeov multiplikátor	-
ρ	multiplikátor pre obmedzenie nerovnosti	-
ε	elektromotorické napätie	V
φ	matica riaditeľnosti	-
ϑ	matica pozorovateľnosti	-
Φ	magnetický tok, charakteristický polynóm systému	Wb, -
$\Phi(\vec{x}_N, N)$	funkcia váhujúca posledný člen na horizonte	-

1 Úvod

Vďaka zložitej spleti rôznych častí ľudského tela a jeho súhre s myšľou, sme schopní vnímať svet okolo nás v jeho úžasnej komplexnosti. Sme schopní odhalovať zákonitosti a následne ich využívať vo svoj prospech. Vedomie, že túto možnosť máme, v nás vytvára neukojiteľnú túžbu po poznaní. Poznanie znova spôsobuje snahu tvoriť, aby fascinujúce myšlienky a objavy neupadli do zabudnutia.

Rovnako ako obraz prírody nie je príroda samotná, tak ani žiadnen iný spôsob opisu ju nie je schopný zachytiť v jej celistvosti tak, ako existuje. Napriek tomu vďaka potrebe človeka poznáť, ľudstvo vymyslelo množstvo spôsobov, ako prostredie okolo nás zachytiť a vytvoriť jeho verný obraz. Tento obraz alebo abstrakcia reality nevie poňať skutočnosť tak komplexne, ako ju vymenáme, ale podľa jej spôsobu zhodenia a "tvaru", vie abstrakcia dokonale kopírovať niektoré jej črty.

V rámci štúdia technických vied sa snažíme ku skutočnému svetu nájsť také abstrakcie, ktoré čo najvernejšie definujú jeho fyzikálne vlastnosti a našu interakciu s ním. Týmito abstrakciami sú najrôznejšie typy modelov a reprezentácií, kde každá reprezentácia odzrkadluje niektoré vlastnosti sveta vhodnejšie. Preto je nevyhnutnou potrebou nie len modely študovať, ale mať aj praktickú skúsenosť s ich vlastnosťami.

Za účelom výučby a spoznávania týchto modelov, vzniklo množstvo najrôznejších zariadení, v rámci ktorých môžeme pozorovať, skúmať, prípade vylepšovať alebo navrhovať nové abstrakcie javov, ktoré tieto zariadenia sprostredkujú. Ide o zariadenia rôznej zložitosti a veľkosti, v závislosti od toho, nakoľko majú komplexné javy reprezentovať.

V odbore teórie riadenia a automatizácie sa jedná o zariadenia reprezentujúce rôzne dynamické systémy, do chodu ktorých je užívateľ schopný zasahovať a prispôsobovať ho svojim predstavám. V poslednom období sú populárnymi systémami v tomto odbore rozličné autonómne vozidlá a drony, nakoľko poskytujú širokú paletu javov, ktorých pochopenie sleduje aj trendy v priemysle, kde autonómnosť mobilných zariadení je jednou z kľúčových otázok. S týmito zariadeniami je ale spätá aj ich cena, odpovedajúca ich komplexnosti a možnostiam, ktoré sa dajú s nimi realizovať.

Okrem autonómnych mobilných zariadení sú pre výskumné účely vyrábané aj špeciálne prístroje, ktorých praktický význam nie je na prvý pohľad celkom zrejmý. Jedná sa o zariadenia reprezentujúce zvyčajne jediný fyzikálny jav, ktorého analýza a riešenie sú využiteľné v širokej skupine nie len priemyselných odvetví, ako súčasť väčšieho systému. Tieto zariadenia často splňajú vysoké nároky na presnosť merania a realizáciu daného fenoménu, nakoľko ich prioritným účelom je odborná analýza javu a modelov využiteľných na interakciu s týmto javom. To sa taktiež prejavuje na cenovej dostupnosti. Medzi takéto zariadenia patria systémy reprezentujúce napr. invertované kyvadlo, aktívne tlmenie vibrácií, balancujúca plocha s loptou a podobne.

S cieľom vytvoriť takéto zariadenia, prioritne na edukačné účely v odbore riadenia, vznikla iniciatíva AutomationShield. V rámci tejto študenti pracujú na projektoch zameraných na návrh

a výrobu pomôcok reprezentujúcich fyzikálne javy, vhodných pre študenta, aby si s ich pomocou osvojil abstrakcie využívané pri inžinierskych činnostach spojených s riadením systémov. Súčasťou tejto filozofie je, aby zariadenia boli jednoduché a finančne nenáročné a tým ich mohol skonštruovať a využívať aj jednotlivec mimo akejkoľvek vzdelávacej inštitúcie. Z tohoto dôvodu sú všetky projekty tejto iniciatívy voľne dostupné v rámci platformy GitHub (viď webová stránka [19]).

Súčasťou existujúcej iniciatívy je set zariadení tzv. shieldov, čo sú malé dosky plošných spojov obsahujúce periférie, reprezentujúce konkrétnie fyzikálne javy. Tie sú kompatibilné s vývojovou doskou Arduino. To obsahuje mikroprocesorovú jednotku od firmy Microchip Technology a poskytuje vlastné periférie, uľahčujúce užívateľovi vytvárať programy, nahrávať ich, prípade komunikovať s ďalšími zariadeniami. Študenti pri práci so shieldom navrhnu a vytvoria program ovládajúci fenomén, ktorý je shieldom sprostredkovany a následne ho nahrajú na Arduino. To je zodpovedné za vykonávanie vytvoreného programu a riadenie sprostredkovaneho fyzikálneho javu.

V rámci projektov už vzniklo viacero shieldov, medzi ktoré patria napríklad:

- FloatShield – Riadenie polohy lopty v uzavretej trubici, nadnášanej prúdom vzduchu z ventilátora nachádzajúceho sa pod ňou (vydaný článok [21]).
- LinkShield – Aktívne tlmenie vibrácií flexibilného člena pomocou servo-motora.
- BlowShield – Riadenie konštantnej veľkosti tlaku v uzavretej nádobe napriek únikom vzduchu.
- BoBShield – Regulácia polohy oceľovej gule v uzavretej trubici, rotujúcej okolo horizontálnej osi.
- HeatShield – Riadenie teploty imitujúce ohrev dýzy dodávajúcej materiál pri 3D tlači (vydaný článok [23]).
- OptoShield – Riadenie intenzity žiarenia svetelného zdroja za pomoci snímania fotorezistorom (vydaný článok [24]).
- MotoShield – Poskytuje rôzne možnosti pre experimenty súvisiace s riadením otáčok motora.

Okrem týchto shieldov sú v procese vývoja aj ďalšie zariadenia. Výsledkom mojej práce na projekte je shield s názvom MagnetoShield. Jedná sa o zariadenie sprostredkujúce riadenie výšky levitácie permanentného magnetu vo vzduchu, kde ho drží sila od elektromagnetu nad ním. Toto zariadenie bolo predmetom mojej bakalárskej práce [15] a bude aj hlavným aktérom mojej diplomovej práce.

2 Vývoj zariadenia MagnetoShield

Shieldy reprezentujú rôzne fyzikálne javy a poskytujú používateľovi interakciu s nimi. Jedným takýmto zariadením je aj MagnetoShield sprostredkujúci jav magnetickej levitácie. Levitácia ako taká, je jav, kedy je celková hybnosť permanentného magnetu vo voľnom priestore nulová $\sum_{i=0}^n \vec{p}_i = 0$, takže sa vznáša v určitom bode bez fyzického kontaktu s okolím. Na takýto magnet pôsobia dve opačne orientované sily, gravitačná sila a magnetická. Magnetická sila je na rozdiel od gravitačnej ovládateľná pomocou elektromagnetu. Používateľom vytvorený riadiaci program vypočíta optimálny elektrický prúd prechádzajúci cievkou elektromagnetu a tým vytvára magnetické pole, ktoré drží magnet vo vzduchu. Z uvedeného vyplýva, že ide o veľmi dynamický jav, ktorý bude potrebné vyhodnotiť mnoho krát za sekundu a nakoľko je magnetická sila mocninovou funkciou, jej prevaha sa v systéme prejaví veľmi rýchlo. Práve tieto vlastnosti systému MagnetoShield ho predurčujú ako výborný fyzikálny jav na testovanie riadiacich algoritmov, ich návrh a celkovo, ako výborný edukačný prostredok za účelom riadenia systému a jeho identifikácie.

Jav magnetickej levitácie je široko študovaný fenomén využívaný v najrôznejších odvetviach priemyslu. Vzhľadom na [12] poznáme 3 typy magnetickej levitácie:

- Elektromagnetické tlmenie (z angl. *electromagnetic suspension*, EMS) – levitácia je spôsobená príťažlivou silou medzi magnetom a elektromagnetom stabilizujúcim polohu levitácie pomocou spätej väzby. O rovnakom type levitácie pojednáva aj táto práca.
- Elektrodynamické tlmenie (z angl. *electrodynamic suspension*, EDS) – jedná sa o princíp založený na odpudivej sile medzi magnetmi. Ak vystavíme vodič meniacemu sa magnetickému poľu, toto pole bude spôsobovať tvorbu napäťia a tým vytvárať vo vodiči elektrický prúd. Ten spôsobuje tvorbu magnetického poľa, opačného k poľu ktoré ho vytvorilo.
- Hybridné elektromagnetické tlmenie – využitie EMS v kombinácii s permanentnými magnetmi vnútri elektromagnetu s cieľom zníženia elektrickej spotreby.

Prvý spomenutý jav je využívaný okrem implementácie vo vlakoch, tzv. *Maglevoch*, ktoré na základe levitácie výrazne znížujú odpor pri pohybe (viac sa ale implementujú princípy EDS), aj pri magnetických ložiskách, alebo pri experimentálnych zariadeniach určených na testovanie a výučbu. Práve tieto experimentálne zariadenia sú predmetom našich najbližších úvah.

Jedná sa o zariadenia pozostávajúce z elektromagnetu, uloženého vo vrchnej časti, pod ktorým je voľný priestor. V ňom sa nachádza objekt z feromagnetického materiálu, prípadne permanentný magnet a ten je účinkami elektromagnetu fahaný silou smerom nahor. Táto sila pomocou spätej väzby vyvažuje gravitačnú silu a teleso v priestore stabilizuje. Tento koncept bol využitý v mnohých vedeckých prácach, ktoré sa líšia predovšetkým spôsobom, ako je spätná väzba vytváraná.

Najjednoduchší spôsob, široko využívaný vzdelenovacími inštitúciami, ale aj hobbystami, je použitie optickej brány. Tu sú zdroj svetla a senzor citlivý na svetlo umiestnené oproti sebe vo

výške, v ktorej má levitácia prebiehať. Levitujúci predmet zakrýva zdroj svetla a tým podáva informáciu o dosiahnutej výške [17]. Tento spôsob však má istú zásadnú nevýhodu, a to tú, že objekt musí byť do výšky levitácie vyzdvihnutý manuálne.

Ďalším spôsobom je použitie optického senzora (napr. ToF senzor, z angl. *Time of Flight*) nachádzajúceho sa pod levitujúcim objektom [13]. Ten meria meniacu sa polohu levitujúceho objektu. Avšak tieto typy senzorov majú minimálnu merateľnú vzdialenosť rádovo v centimetroch a nevedia merať polohu v bezprostrednej blízkosti.

Spôsob, ktorý využíva levitátor MagnetoShield, je meranie polohy za pomocí Hallovho senzora, ktorý deteguje prítomnosť magnetického poľa a vie určiť aj jeho intenzitu. Aj toto riešenie podlieha určitým nevýhodám a to tomu, že Hallov senzor meria celkové magnetické pole, teda aj to, ktoré je generované elektromagnetom a je premenlivé. Z tohto dôvodu som sa pri návrhu MagnetoShieldu rozhodol Hallov senzor umiestniť do dolnej časti levitátora, pod levitujúci objekt (minimalizoval som vplyv elektromagnetu na meranie). Taktiež som využil silný permanentný neodymový magnet, ktorého magnetické pole je násobne väčšie ako pole elektromagnetu (dominantná a nemenná zložka poľa je pomerne jednoducho merateľná a vzhladom k jej veľkosti je vplyv elektromagnetu na Hallov senzor, uložený v dolnej časti, možné zanedbať).

2.1 MagnetoShield R1

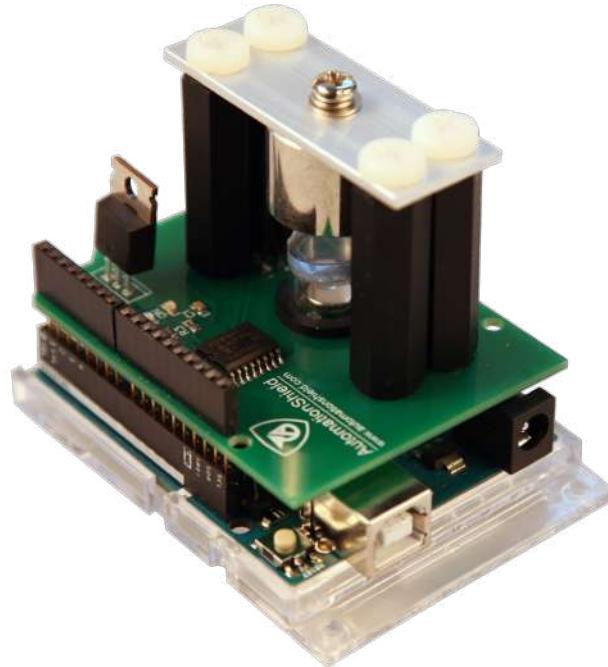
Ako som spomíнал, vývoju MagnetoShieldu bola venovaná moja bakalárska práca [15], ktorej ústredným cieľom bol návrh hardvérovej časti, realizácia návrhu (Obr. 2.1) a následné vytvorenie jednoduchého obvodu riadenia, potvrdzujúceho funkčnosť zariadenia. Pre rozlíšenie od ďalších verzií, ktoré boli navrhnuté následne, použijem označenie MagnetoShield R1 pre prvotný návrh (R z angl. *release* = vydanie/prevedenie).

Po hardvérovej stránke išlo o veľmi jednoduchý elektrický obvod (Obr. 2.2), pozostávajúci z elektromagnetu, tranzistoru riadeného poľom s hradlovou oxidovou vrstvou (ďalej MOSFET – z angl. *metal-oxide-semiconductor field-effect transistor*), digitálno-analógového prevodníka (ďalej DAC prevodník), Hallovho senzora a dodatočných pasívnych komponentov ([15, Kap. 2, Kap. 3]). Hallov senzor snímal intenzitu magnetického poľa z premanentného magnetu levitujúceho nad ním. Na základe získaných informácií sa dalo určiť, v akej pozícii sa permanentný magnet nachádza. Následne riadiaci systém v podobe mikroradiča (nie je súčasťou MagnetoShiledu ale rozhrania Arduino, pre ktoré je shield vyrobený) vypočítal vhodný vstup v podobe napäťia. Informácia o výpočte sa poslala na DAC prevodníka (súčasť MagnetoShieldu) a ten generoval napätie 0 až 5 V. To bolo privedené na MOSFET, ktorý na jeho základe vytváral premenlivé napätie na elektromagnete 0 až 12 V a tým menil požadovanú hodnotu elektrického prúdu generujúceho magnetické pole.

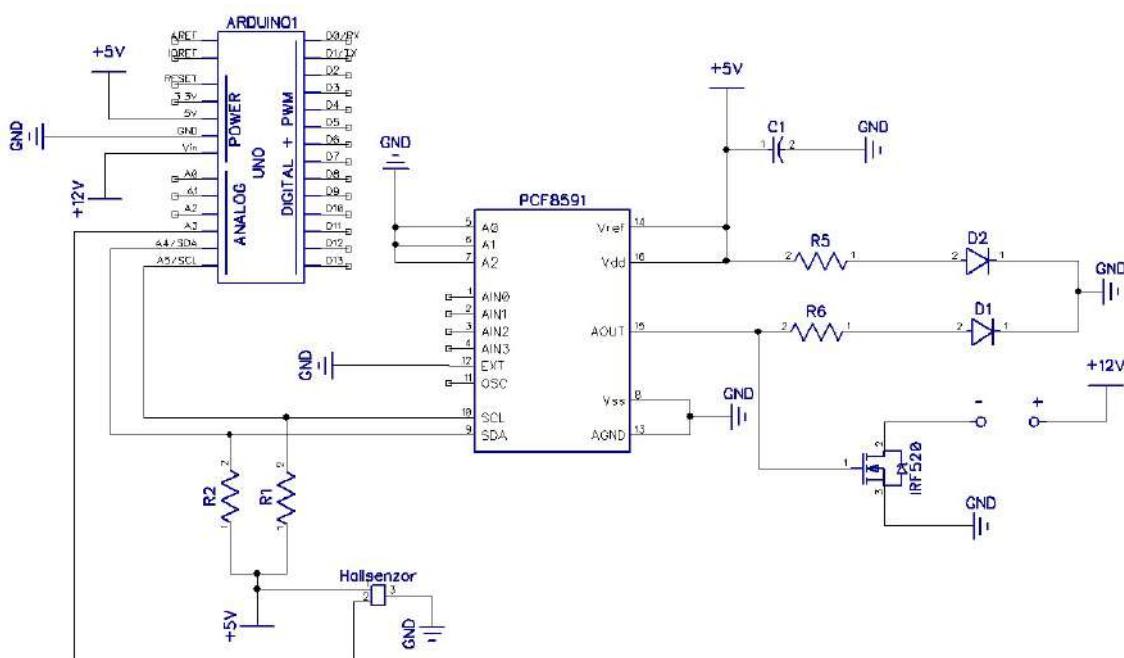
Ako je vidno, MOSFET je použitý ako spojovací prvok medzi "malým" riadiacim napäťom (5 V) a "veľkým" napájacím napäťom (12 V), ktoré je potrebné regulovať (Obr. 2.2 – kladná a záporná svorka za 12 V napájaním v schéme označujú miesto pripojenia elektromagnetu).

Okrem návrhu elektrického obvodu, bakalárska práca tiež opisovala návrh a osadenie ostatných súčastí tvoriacich zariadenie, ktoré však nepatrili do elektrického obvodu. V tomto zmysle bol uvedený detailný opis pasívnych komponentov ako stĺpkov, skrutiek a iných častí konštrukcie, ktoré mali za úlohu pridržať elektromagnet vo vrchnej pozícii a zároveň zabrániť, aby sa permanentný magnet trvalo prílnul ku elektromagnetu (sila medzi magnetom a oceľovým obalom elektromagnetu je od určitej vzdialenosť väčšia ako gravitačná sila).

Softvérové rozhranie pozostávalo z knižnice naprogramovanej v jazyku C/C++ pre vývojové prostredie Arduino IDE. To ponúkalo balík funkcií určených na komunikáciu medzi prototypizač-



Obr. 2.1: MagnetoShield R1.



Obr. 2.2: Elektrický obvod zariadenia MagnetoShield R1.

nou doskou Arduino a MagnetoShieldom [15, Kap. 4]. S jeho pomocou bol používateľ schopný vytvárať riadiace programy pre levitáciu, založené na jednoduchej spätej väzbe, ktorá informovala o polohe vznášajúceho sa magnetu. Požívateľ mal dosah na DAC prevodník, ktorému vedel zadať hodnotu napäťia, Hallov senzor z ktorého sa číitali dátu o intenzite magnetického poľa a teda aj o polohe magnetu, nastavenie požadovanej výšky levitácie a funkciu kalibrácia. Táto funkcia zisťovala napätie na Hallovom senzore v prípade, že sa magnet nachádza v dolnej pozícii, teda nelietal, a v prípade, že sa nachádza v najvyššej možnej pozícii. Na základe tohto merania sa vytvoril vzorec na prepočet napäťia zo senzora na vyjadrenie pozicie, kde sa magnet nachádzal. Tento vzťah bol lineárny a rozmer výstupu boli percentá. Čo sa týka DAC prevodníka, vytvorená funkcia sprostredkúvala I²C komunikáciu medzi Arduinom a prevodníkom. DAC prevodník použitý v MagnetoShield R1 (taktiež použitý v MagnetoShield R2) bol 8-bitový čip PCF8591. To znamená, že riadiaci systém (prototypizačná doska Arduino) posielal prevodníku správu v podobe 8-bitového čísla, na základe ktorého prevodník generuje príslušnú hodnotu napäťia v rozmedzí 0 až 5 V. Toto 8-bitové číslo, ktoré nadobúda hodnoty 0 až 255 a mapuje sa do intervalu 0 až 5 V, budem ďalej nazývať DAC level.

V závere práce som uviedol ukážkový príklad použitia zariadenia MagnetoShield R1, pri implementácii PID regulátora, ako riadiaceho algoritmu pre levitáciu [15, Kap. 5]. V rámci príkladu som ukázal vplyv jednotlivých zložiek regulátora, uviedol som hodnoty naladeného regulátora a prezentoval som výsledky. V nich som uvádzal požadovanú výšku levitácie v percentách (percentuálne vyjadrenie v akej výške má magnet lietať, ak najvyššia pozícia predstavuje 100 %), zmeranú výšku Hallovým senzorom a vstupy generované PID regulátorom. Výsledkom tohto príkladu bolo úspešné prevedenie javu magnetickej levitácie permanentného magnetu. No napriek vydarenému experimentu ostávalo niekoľko nedokonalostí, ktoré sa s MagnetoShieldom R1 spájali a niektoré z nich som uviedol aj v závere svojej práce [15, str. 44-45].

- V prvom rade som konštatoval, že sice levitácia prebieha a možno ju považovať za ustálenú, no napriek tomu dochádza k určitej oscilácií okolo požadovanej hodnoty. Uviedol som niekoľko možných dôvodov, ktoré mohli tento jav spôsobovať, ako napríklad nehomogénnosť magnetického poľa v horizontálnom smere (ak sa magnet vychýli z osi nad halovým senzorom do strany, regulátor to vyhodnotí ako zmenu výšky magnetu) alebo neschopnosťou PID riadenia takýto systém uregulovať.
- Taktiež hardvérové riešenie bolo uspôsobené tak, aby bolo kompatibilné s prototypizačnou doskou Arduino UNO a na základe toho som pre I²C komunikáciu zvolil piny A4 a A5. Avšak takáto konfigurácia je použiteľná iba na doske Arduino UNO. U ostatných dosiek sa piny na I²C komunikáciu nachádzajú inde.
- Posledným bodom bola navrhovaná zmena v dizajne bránky pridŕžajúcej elektromagnet, nakoľko pôvodný dizajn bol nepotrebné robustný a vyžadoval veľké množstvo materiálu.

Po neskoršej dôslednej analýze sa zistilo, že hlavným dôvodom oscilácie nebola nedostatočná sofistikovanosť riadiaceho PID algoritmu alebo nehomogénnosť poľa (aj keď to malo isté degradáčné účinky na priebeh levitácie), ale nedokonalosť v elektrickom návrhu hardvéru. Riešením stanovených problémov mala byť nová verzia MagnetoShieldu, R2.

2.2 MagnetoShield R2

Prvotná verzia MagnetoShieldu bola sice funkčná, ale mala určité nedostatky, ktoré boli pomerne ľahko identifikovateľné, čo viedlo k nápadu vytvoriť novú verziu, v ktorej budú nájdené nedokonalosti odlaďené. Výsledkom tejto iniciatívy bol vznik nového návrhu MagnetoShield R2 (Obr.

2.3), ktorý okrem nápravy spomenutých chýb mal umožniť aj nové možnosti v rámci riadenia, identifikácie a implementácie algoritmov na daný systém. Za väčšinu zmien a získaných poznatkov z návrhu tejto verzie som vďačný prof. Ing. Gergelymu Takácsovi, PhD. [20].



Obr. 2.3: MagnetoShield R2.

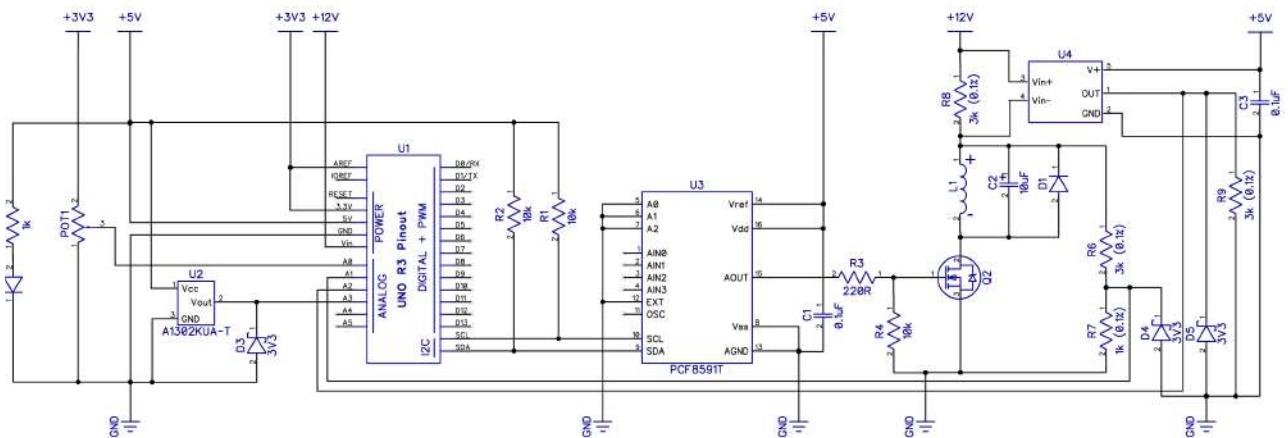
Prvou zásadnou zmenou bola úprava návrhu hardvéru pridaním niekoľkých komponentov (Obr. 2.4). Prvým takýmto komponentom bol kondenzátor **C2**. Jeho účelom bolo vyrovnávať prechodové javy na elektromagnete v prípade zmeny elektrického prúdu¹. Okrem kondenzátora sa do schémy pridala aj ochranná dióda **D1** ako ochrana proti spätnému prúdu, aby nepoškodil MOSFET a prípadne aj zdroj napäťia.

Ďalšou výraznou hardvérovou zmenou bolo pridanie senzoru **U4** na meranie elektrického prúdu na elektromagnete, meranie napäťia na elektromagnete pomocou napäťovej deličky **R6**, **R7** a pridanie potenciometra **POT1** ako možnej analógovej referencie.

Tu sa dostávame k otázke, nie len ako opraviť chyby, ktoré vznikli pri návrhu MagnetoShield R1, ale aj ako vylepšiť samotný potenciál zariadenia. Spomenuté hardvérové rozšírenia boli pridané práve za týmto účelom. Predošlá verzia MagnetoShieldu pracovala s jednoduchou spätnou väzbou, ktorá nám vracala iba informáciu o momentálnej polohe levitujúceho magnetu. Následne spracovávala regulačnú odchyľku, ako jedinú informáciu o procese levitácie. Táto informácia je pre samotný jav postačujúca, avšak si treba uvedomiť, že poloha nie je jediným stavom zabezpečujúcim levitáciu. Ako som v úvode spomenul, pre ustálenú levitáciu je potrebné mať nulovú celkovú hybnosť sústavy. Čiže ak máme informáciu o rýchlosťi, určite nám to pomôže pri generovaní správneho stabilizujúceho vstupu. Naviac poloha je regulovaná na základe výstupu z elektrického obvodu, ktorý má taktiež svoju dynamiku, ktorú nie je možné zanedbať. Takže vstupný signál je tým kvalitnejší, čím viac kľúčových veličín sa podieľa na jeho tvorbe.

Taktiež je nutné poznamenať, že sa nejedná o lineárny systém. Síce nelineárne riadenie do-

¹Zmenou napájacieho napäťia sa zmení elektrický prúd a nakoľko cievka v elektromagnete má istú vlastnú indukčnosť, vytvára vlastné indukované napätie podľa $U(t) = L \frac{dI}{dt}$. Cieľom kondenzátora malo byť tieto javy vyvážovať.



Obr. 2.4: Elektricky obvod zariadenia MagnetoShield R2.

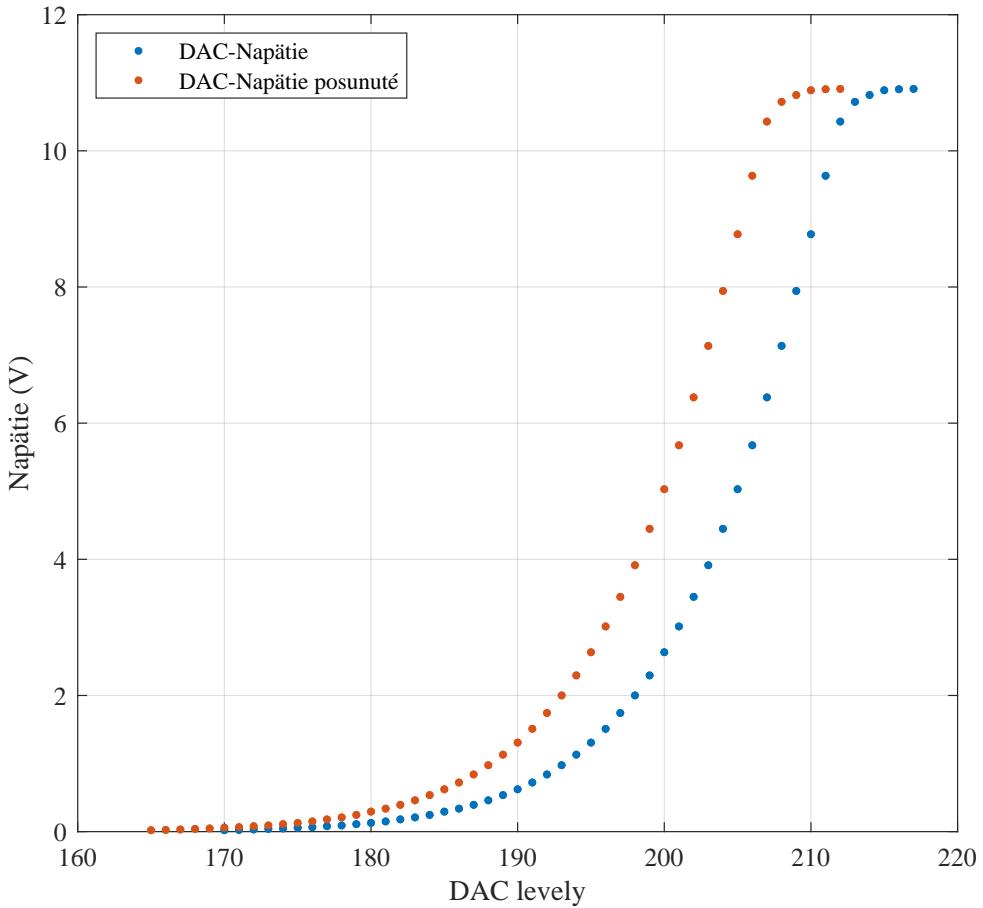
teraz na MagnetoShield implementované nebolo, ale aby sme si vôbec sprístupnili túto možnosť, je nutné mať opis systému, ktorý presne definuje, kde a v akom rozsahu sa nelineárnosti nachádzajú. Túto v skutočnosti veľmi komplexnú informáciu nie sme schopní zaobaliť do späťnej väzby vracajúcej regulačnú odchýlku polohy.

Z tohto dôvodu musíme vedieť vytvoriť nie len naladený regulátor, ale aj určitý systém, ktorý bude brať ohľad na uvedené fakty. Práve k tomuto účelu boli do systému pridané komponenty na meranie napäcia a prúdu na elektromagnete. Jedna z možností, ako zohľadniť chovanie sa systému počas regulácie, je vytvorenie jeho modelu napríklad v stavovom priestore. Témam modelovania a práci s modelom sa však budem venovať až v ďalších kapitolách, Kap. 5 a Kap. 6.

MagnetoShield R2 dostal niekoľko vylepšení aj v rámci softvérového rozhrania. Knižnica použitá pre predošlú verziu bola obohatená o funkcie spracúvajúce dátu z pridaných senzorov napäcia a prúdu a taktiež analógovej referencie z potenciometra. S uvedením nových senzorov a začiatkom uvažovania o modeli systému je tiež spojený spôsob, ako interpretovať nameranú pozíciu levitujúceho magnetu. Zatiaľ čo pri PID riadení bola informácia o polohe relatívna hodnota, pri použití stavového priestoru potrebujeme, aby táto informácia mala určitý absolútny charakter. To je potrebné práve z dôvodu, že systém sa so vzdialenosťou mení a my potrebujeme túto meniacu sa veličinu nejakým spôsobom opísť a priradiť jej vlastnosti pomocou funkcionálnej závislosti. Z tohto dôvodu bolo meranie polohy rozšírené o balík na seba nadväzujúcich funkcií umožňujúcich prepočet nameraného napäcia na Hallovej sonde do Gaussov (jednotka intenzity magnetického poľa) a na vzdialenosť permanentného magnetu v milimetroch.

Treba zdôrazniť, že po spomenutých úpravách ako hardvéru, tak aj softvéru sa podarilo dosiahnuť požadované a očakávané výsledky. To znamená, ustálená levitácia magnetu v priestore, bez oscilácie okolo žiadanej hodnoty, pričom bol použitý regulátor založený na opise systému v stavovom priestore. Napriek tomuto úspechu sa po zhotovení viacerých kusov MagnetoShieldu R2 vynoril opäť jeden problém.

Ako bolo spomenuté v Kap. 2.1, systém je ovládaný pomocou DAC prevodníka, ktorý prostredníctvom MOSFETu upravuje "veľké" napätie. MOSFET je však konštrukčne navrhnutý v prvom rade ako spínací prvok. Jeho použitie spôsobovalo jav, že ak sme privádzali na elektródu



Obr. 2.5: Závislosť napäcia od DAC levelov na elektromagnete.

G (z angl. *gate*)² napätie 0 V až približne 3.5 V (180 DAC levelov, vid’ Obr. 2.5), MOSFET ostával zatvorený, teda neprepúšľal medzi elektródami D (z angl. *drain*) a S (z angl. *source*) žiadnen prúd a napätie na elektromagnete bolo 0 V. Potom v rozmedzí 3.5 V až 4.2 V sa choval práve tak, ako bolo potrebné, teda s rastúcim napäťom, charakterizovaným DAC levelmi, privádzaným na G, rástol aj prúd tečúci cez D a S a teda aj napätie na elektromagnete. Následne, vyššie napätie ako 4.2 V (DAC levely 212 a vyššie) už nespôsobovalo žiadnu zmenu. Na Obr. 2.5 modrou, môžeme vidieť túto závislosť vyjadrenú graficky, kde na x-ovej osi máme znázornené hodnoty DAC levelov, ktoré posielalo Arduino na DAC prevodník a následne na elektródu G a na y-ovej osi sú hodnoty napäcia na elektromagnete³. Takže sice máme k dispozícii potenciálne 256 rôznych napäťových úrovní na reguláciu levitácie, ale v skutočnosti chovanie MOSFETu dovoľuje využitie iba pomerne malej časti tohto intervalu a teda “pracovné pásmo”, ktoré vieme použiť je dosť úzke.

Sám o sebe by to neboli príliš veľký problém, samozrejme znižujú sa naše možnosti pre riadenie daného javu, ale principiálne to neobmedzuje funkčnosť systému ako takého. Skutočným problémom bolo porovnanie tejto DAC-napäťovej charakteristiky pri dvoch rozdielnych Magne-

²MOSFET má 3 elektródy a to G, D a S.

³Prevod medzi DAC levelom U_{DAC} a napäťom U_{out} , ktoré privádzame na G pomocou 8-bitového čísla sa dá opísť ako $U_{out} = \frac{5V}{255} U_{DAC}$.

toShieldoch. Zistilo sa, že síce sa jedná o dva MOSFETy toho istého typu od toho istého výrobcu, no nakoľko toto pracovné pásmo je v MOSFETe veľmi úzke, pri každom modeli môže byť jemne posunuté. Takýto posun, i keď len o 5 DAC levelov výrazne mení chovanie sa systému voči očakávaniu (viď. Obr. 2.5 červená – posunutý graf).

Táto rozdielnosť jednotlivých MOSFETov by nebola neprekonateľnou prekážkou. No na vrhnutie a vytvorenie univerzálneho softvérového rozhrania, ktoré by bolo schopné prepočítať posun grafu a aj jeho prípadné transformácie, by bolo pomerne náročné a výpočtovo komplikované. Navyše by toto riešenie nezmenilo nič na tom, že pracovný interval je veľmi úzky a väčšina potenciálu na riadenie ostáva nevyužitá. Z týchto dôvodov sa pristúpilo k ďalšiemu prerobeniu hardvéru a z časti aj softvéru, ktorého výsledkom je MagnetoShield R3.

2.3 MagnetoShield R3

Spomenuté nedostatky si opäť vyžadovali vytvorenie novej verzie, v rámci ktorej nastala pomerne rozsiahla zmena hardvéru aj softvéru. Tak vznikla “pracovná” verzia MagnetoShield R3 na ktorej boli v priebehu času odladené a odstránené nedostatky. Tieto zmeny postupne konvergovali k výslednému a konečnému produktu, ktorý sa na rozoznanie od pracovnej verzie označil ako MagnetoShield R4.

Nakoľko výsledné úpravy zapríčinili nové označenie zariadenia, popis vykonaných zmien a aj kompletný opis produktu je uvedený v nasledujúcej kapitole.

3 MagnetoShield R4 – Hardvér

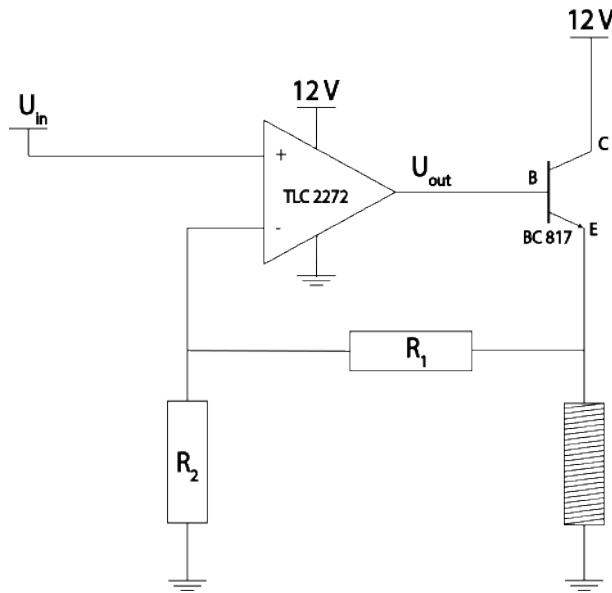
Na pracovnej verzii MagnetoShield R3 bolo realizovaných niekoľko hardvérových zmien, ktoré viedli k finálnej podobe MagnetoShieldu R4 (Obr. 3.1). Najzásadnejšia nastala v rámci hardvéru v tom, ako pomocou “malého” riadiaceho napäťa ovládame “veľké” napájacie napätie na elektromagnete. Doteraz sme na tento účel využívali MOSFET, ale ako bolo spomenuté, tento komponent nie je celkom vhodný na spojité reguláciu napäťa.



Obr. 3.1: MagnetoShield R4.

V rámci výskumu a testovania pomocou platformy *Ltspice*, ktorá slúži na simuláciu elektrických obvodov, som dospel k záveru, že najvhodnejším riešením bude použitie obyčajného NPN tranzistora (BC817) spoločne s operačným zosilňovačom (TLC2272) v neinvertujúcom zapojení zobrazrenom na Obr. 3.2.

Jedná sa o schému v ktorej DAC prevodník posiela zvolené napäťové hodnoty najskôr do operačného zosilňovača (ďalej len opamp – z angl. *operational amplifier*), úlohou ktorého je vstupný signál U_{in} zosilniť. Opamp má napájanie 12 V a v jeho špecifikácii sa uvádzá že to je rail-to-rail. Inými slovami, ak je opamp napájaný určitým napäťom, rozsah hodnôt ktoré môže výstupný signál U_{out} nadobúdať, bude rovnaký ako toto napájacie napätie (v praxi je maximálny výstup trošku menší ako napájanie). Zosilnený signál ďalej putuje na tranzistor, kde je privádzaný



Obr. 3.2: Schéma zosilňovača na MagnetoShield R4.

na pin B nazývaní báza (z angl. *base*). Princíp fungovania tu je skoro rovnaký ako pri MOSFETe. Na bázu sa privádzajú signál (riadiaci) a ten prepúšťa iný signál (riadený), ktorý sa nachádza medzi emitorom E (z angl. *emitter*) a kolektorom C (z angl. *collector*). Okrem tejto podobnosti toto zapojenie obsahuje aj niekoľko rozdielov. Prvý rozdiel je v tom, že na MOSFET sme privádzali malý signál maximálne 5 V, tu je na báze zosilnený veľký signál s maximálnou hodnotou 12 V. Druhým rozdielom je to, že súčasťou hovoríme stále o napäti, nakoľko je veličinou na ktorú máme priamo dosah, ale tranzistor v skutočnosti funguje na prenose prúdových signálov. Tým, že zvýšim výstupné napätie z opampa, zvýšim aj výstupný prúd. Ten je privádzaný na bázu a "pomáha" prúdu medzi C a E preraziť a tým zvýšiť napätie a v konečnom dôsledku prúd na elektromagnete. Prúd je aj dôvodom prečo na reguláciu nie je dobré použiť iba opamp bez tranzistora (aj toto riešenie je možné). Opamp má obmedzený prúd, ktorý je schopný poskytnúť na výstupe a my nevieme presne, aký veľký bude odber cievky elektromagnetu, preto opamp iba ovláda prieplustnosť tranzistoru a prúd pre cievku je ľahko z napájacieho zdroja.

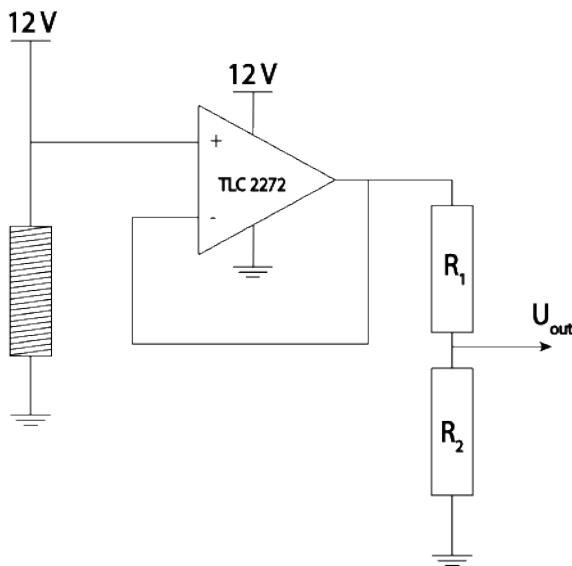
Na Obr. 3.2 vidíme, že operačný zosilňovač má okrem vstupného signálu označeného ako "+" aj vstup pre signál "-". Tento vstup poskytuje opampu množstvo rôznych spôsobov zapojenia. U nás, ako som už spomíнал, bolo použité neinvertujúce zapojenie. To hovorí o tom, že výstupný signál nemôže byť menší/rovný ako vstupný (ak je vstup kladné napätie). Toto tvrdenie možno vyjadriť rovnicou

$$U_{\text{out}} = (1 + A)U_{\text{in}}. \quad (3.1)$$

Premenná A v rovniči predstavuje pomer odporov R_1 a R_2 . Tým pádom prenos opampu G , tiež nazývaný zosilnenie (angl. gain) možno zapísat nasledovne:

$$G = \frac{U_{\text{out}}}{U_{\text{in}}} = 1 + \frac{R_1}{R_2} = \frac{R_1 + R_2}{R_2}. \quad (3.2)$$

Pomocou správne zvolených odporov R_1 a R_2 teda vieme nastaviť zosilnenie tak, aby sme pri vstupnom napäti 0 V dostali na výstupe 0 V a pri vstupe 5 V dostali na výstupe 12 V. To dosiahneme tak, že si zvolíme jeden z odporov, v mojom prípade R_2 , a iba dopočítame R_1 .



Obr. 3.3: Schéma merania napäťia na elektromagnete pre MagnetoShield R4.

$$U_{in} = 5 \text{ V}, \quad (3.3)$$

$$U_{out} = 12 \text{ V}, \quad (3.4)$$

$$R_2 = 4.7 \Omega, \quad (3.5)$$

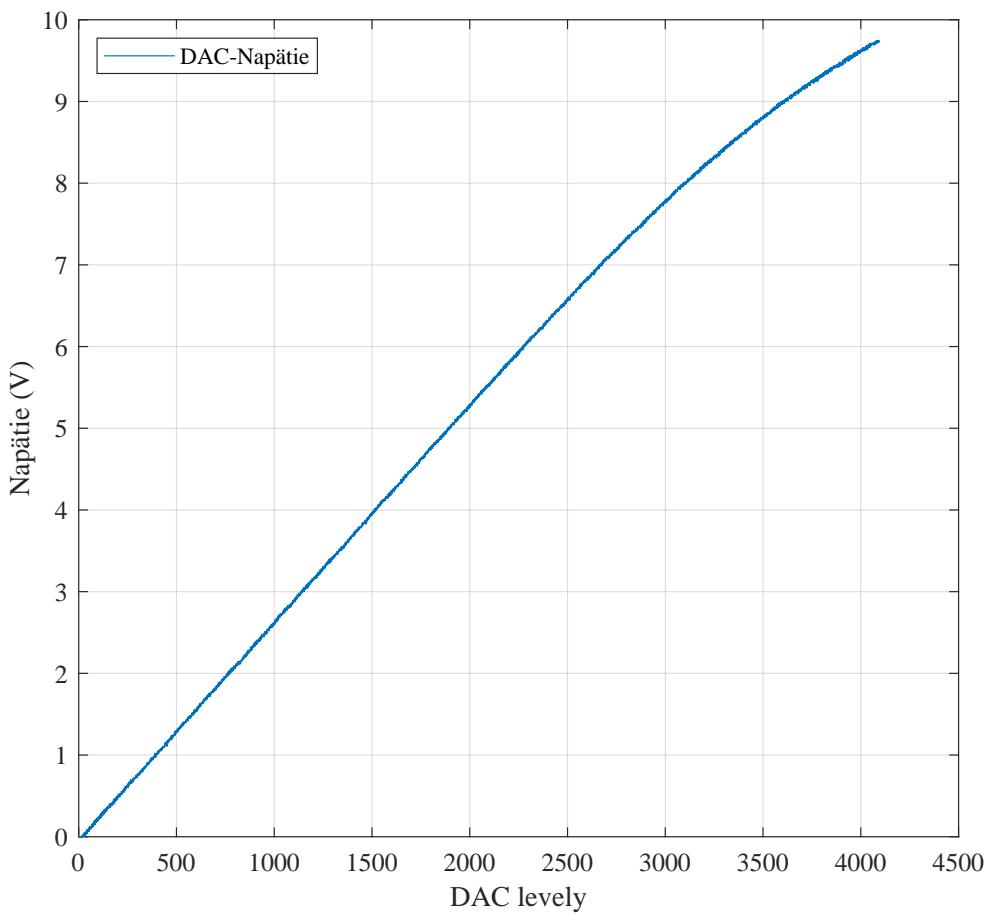
$$\frac{12}{5} = 1 + \frac{R_1}{4.7}, \quad (3.6)$$

$$R_1 = \left(\frac{12}{5} - 1\right)4.7 = 6.58 \Omega. \quad (3.7)$$

Uvedené hodnoty odporov boli zakomponované aj do návrhu MagnetoShield R4 s tým rozdielom, že odpor R_1 je reprezentovaný nastaviteľným potenciometrom **R10** na Obr. 3.5, takže prípadné rozdielnosti v chovaní sa komponentov v obvode sa dajú vykompenzovať (hodnota odporov sa udáva $\pm 10\%$ pre klasické, $\pm 1\%$ pre presné).

Druhá zmena v obvode nastala pri meraní napäťia na elektromagnete. Ako si možno všimnúť na Obr. 2.4 v Kap. 2.2 napätie na elektromagnete sa meria pomocou napäťovej deličky z odporov **R6** a **R7**. Toto riešenie sice vracia informáciu o napätií, ale je skreslené vplyvom odberu prúdu cez spomenuté oditory (pridanie nového elektrického prvku do obvodu vždy ovplyvní napäťové a prúdové pomery v celom obvode). Nové riešenie v sebe zahŕňa opäť použitie opampu. Nakoľko opamp má na vstupe veľkú (ideálne nekonečnú) impedanciu, jeho prúdový odber na vstupe je minimálny a tým pádom nedochádza ku skreslovaniu napäťových hodnôt privádzaných na vstup. Ak teda dokážeme nastaviť prenos opampu $G = 1$ dostaneme na jeho výstupe rovnaké napätie, aké sa nachádza na vstupe a teda aj na elektromagnete. Zapojenie vytvárajúce jednotkové zo-silnenie predstavuje jednoduchá spätná väzba zobrazená na Obr. 3.3. Za opampom sa nachádza napäťová delička s hodnotami odporov $R_1 = 3 \text{ k}\Omega$ a $R_2 = 1 \text{ k}\Omega$. Tá zabezpečuje mapovanie napäťia na elektromagnete do hodnôt, spracovateľných Arduinom prostredníctvom analógových vstupov. Maximálna hodnota vstupného napäťia pri analógovom spracovaní signálu je v prípade projektu MagnetoShield nastavená na 3.3 V (može sa týkať aj ostatných projektov iniciatívy AutomationShield). Toto riešenie je v schéme zapojenia na Obr. 3.5 znázornené odormi **R6**, **R7** a opampom **U5.2**.

Výsledkom implementácie opampov do obvodu je teda linearizácia vzťahu medzi napäťím



Obr. 3.4: Závislosť napäcia na elektromagnete od DAC levelov.

generovaným výpočtovou jednotkou – Arduinom, v podobe DAC levelov a napäťím meraným na elektromagnete¹. Týmto spôsobom sa podarilo odstrániť problém s úzkym pracovným pásmom MOSFETu znázorneným na Obr. 2.5 a taktiež problém s rozlišným chovaním sa komponentov. Pre porovnanie výsledný pomer DAC levelov a napäťia na elektromagnete je znázornený na Obr. 3.4 (rozdiel v rozsahu DAC levelov je daný použitím iného DAC prevodníka). Táto skutočnosť výrazne zjednodušuje a rozširuje dosah riadiaceho algoritmu na akčný člen.

Taktiež sa počas vytvárania nového modelu systému zistilo, že spomenutý kondenzátor **C2**, ktorý je súčasťou obvodu na Obr. 2.4, neplní v obvode takú funkciu s akou bol pridaný. Totiž našou snahou je dynamicky meniť napätie a na základe toho aj elektrický prúd na elektromagnete. Pridanie dodatočného kondenzátora spôsobovalo efekt podobný tlmeniu, teda spomaľoval dynamiku s akou sme boli schopní ovládať elektromagnet. Taktiež by bolo nutné dynamiku kondenzátora zahrnúť do rovnice opisujúcej elektrické pomery v obvode, čo by celý problém jedine komplikovalo bez výslednej pridanej hodnoty.

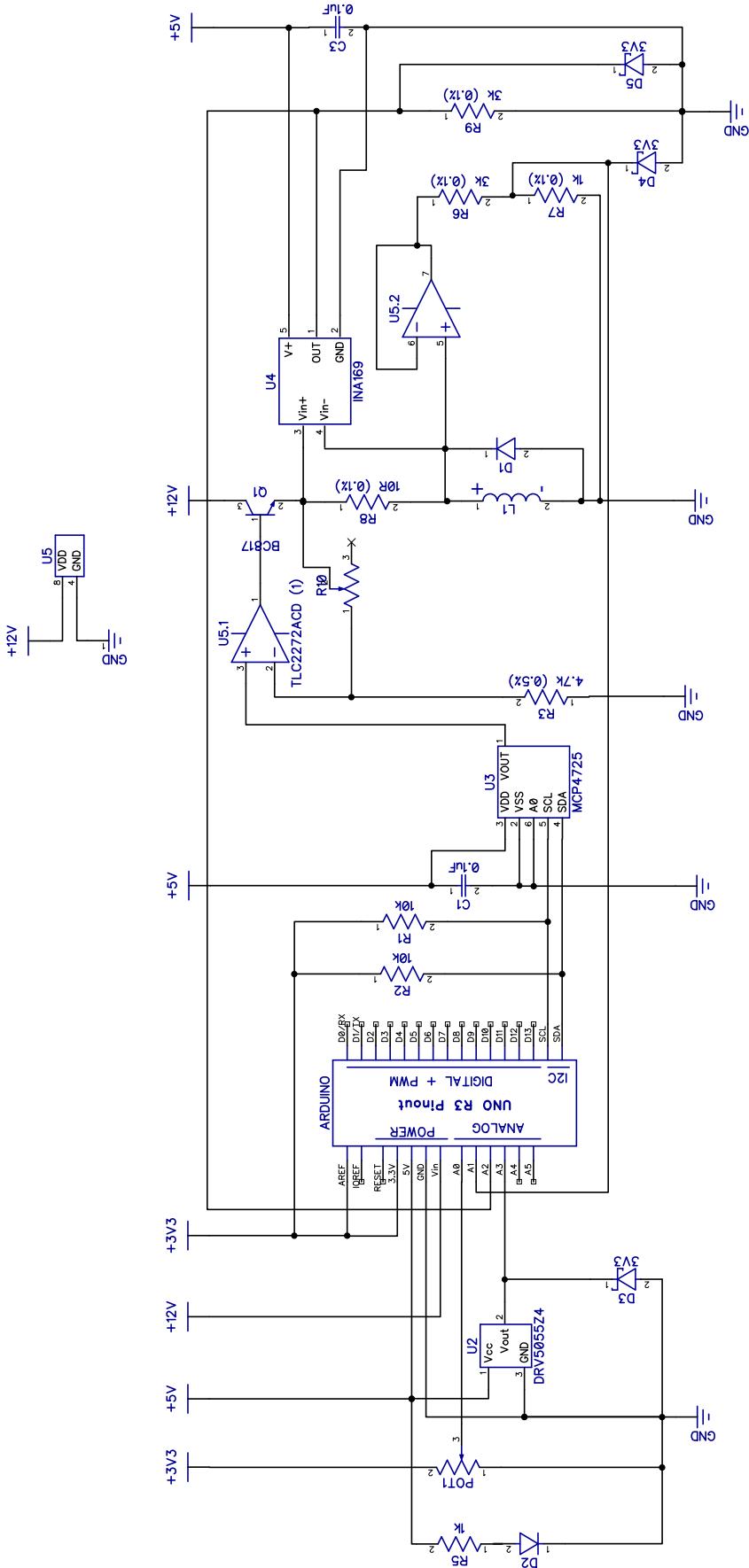
Mimo týchto hardvérových zmien došlo taktiež k výmene Hallovho senzora. Teraz využívaný senzor DRV5055Z4 má takmer totožné vlastnosti ako predtým použitý A1302KUA-T. Výmena senzora nastala z dôvodu zastavenia výroby senzoru A1302KUA-T.

Posledná hardvérová zmena, ktorá sa realizovala na MagnetoShield R4 bola výmena 8-bitového DAC prevodníka PCF8591 za 12-bitový MCP4725. To poskytuje riadenie “malého” napäťia generovaného DAC prevodníkom s oveľa vyšším rozlišením a tým pádom aj regulácia napäťia na elektromagnete môže byť precíznejšia². Výsledná schéma elektrického obvodu pre MagnetoShield R4 je na Obr. 3.5. Zmena DAC prevodníka v sebe ale nesie taktiež úpravu softvérového rozhrania, ale to je predmetom až nasledujúcej kapitoly. Na záver uvádzam v tabuľke 3.1 zoznam komponentov, z ktorých hardvér MagnetoShiledu R4 pozostáva³.

¹Vzťah je len približne lineárny a tak funkcie sprostredkúvajúce túto konverziu v softvérovej časti, používajú polynomiálne vyjadrenie prevodu. Uvažovanie o tomto vzťahu ako o lineárnej funkcií však nemá výraznejší dopad na chod riadenia.

²Rozlíšenie 8-bitového prevodníka je 1/1024 V na DAC level, a rozlíšenie 12-bitového je 1/4096 V na DAC level.

³Sú tu uvedené dva permanentné magnety. V mojej práci som pre účely riadenia a identifikácie využíval prvý z uvedených magnetov. Na stránke projektu [19] bol za účelom riadenia a identifikácie použitý zvyčajne druhý uvedený magnet.



Obr. 3.5: Schéma elektrického obvodu MagnetoShield R4

Názov komponentu	Označenie v schéme Obr. 3.5	Označenie komponentu
Doska plošných spojov (PCB)	-	-
Prúdový senzor	U4	INA169NA/250
Opamp	U5.1,U5.2	TLC2272ACD
DAC prevodník	U3	MCP4725A0T-E/CH
Keramický kondenzátor $0.1\mu F$	C1,C3	C0805C104K5RAC7411
Dióda DO214AC	D1	BYG20D-E3/TR3
Led dióda červená 20mA 0805	D2	150080RS75000
Zernerová dióda 3v3 SOD323	D3,D4,D5	MM3Z3V3ST1G
Potenciometer	POT1	CA9MV 10K
Tranzistor NPN	Q1	BC817K-16HR
Odpór 10k 0805	R1,R2	RT0805FRE0710KL
Odpór 4.7k (0.5%) 0805	R3	ERA-6AED472V
Odpór 1k 0805	R5	RR1220P-102-D
Odpór 3k (0.1%) 0805	R6,R9	RT0805BRD073KL
Odpór 1k (0.1%) 0805	R7	PCF0805R-1K0BI
Odpór 10R (0.1%) 0805	R8	CPF0805B10RE
Potenciometer 10k	R10	3214W-1-103E
Hallov senzor	U2	DRV5055Z4 (A1302KUA-T)
Elektromagnet	L1	P20 x 15 24V 2.5kg
Magnet	-	Neodyne 8x2mm N38 alebo Neodyne 9x2mm N35
Plastová trubica	-	BP-NCCLT12AC-L500
Piny na pripojenie	-	PRT-10007
Plastová skrutka (dolná)	-	50M050080P010
Plastová skrutka (horná)	-	50M040070D006
3D tlač	-	-

Tabuľka 3.1: Hardvérové komponenty zariadenia MagnetoShield.

4 MagnetoShield R4 – Softvérové rozhrania

MagnetoShield R4 je už pomerne komplikované zariadenie, ak si ho porovnáme s jeho prvou verziou. Ponúka veľa nových funkcií a možností na prácu so systémom levitujúceho magnetu. Informáciu o systéme už nereprezentuje iba číslo opisujúce, kde sa nachádza levitujúci magnet v priestore, ale máme dosah aj na vnútorné veličiny, ako sú napätie a prúd tečúci cez elektromagnet. Taktiež môžeme do systému zasahovať aj priamo pomocou potenciometra (analógová referencia) a po napísaní vhodného kódu, môžeme jeho pomocou ovládať napríklad požadovanú výšku levitácie, alebo ručne meniť veľkosť napäťia na elektromagnete a vidieť, aký účinok má meniac sa napätie na vytváranie magnetickej sily medzi magnetmi.

Na to, aby sa so zariadením dalo jednoducho manipulovať a všetky tieto informácie vedel používateľ ľahko získať podľa potreby, bolo vyvinutých niekoľko aplikačných rozhraní. Tie využívajú rôzne programovacie jazyky a ich kombinácie tak, aby používateľovi, podľa možností, čo najviac zjednodušili prístup k systému a vytvárali základnú paletu funkcií a programov, pomocou ktorých je možné vytvárať zložitejšie a komplexnejšie typy riadenia. Pre podrobný opis a zdrojové kódy knižníc a ich častí, viď stránku projektu [19].

Nakoľko idem hovoriť o softvéri a jeho priamych dôsledkoch na systém, je nutné spomenúť ešte jeden zásadný poznatok. V schéme na Obr. 3.5 sú v rámci “veľkého” okruhu viaceré elektrické komponenty. Každý z nich spôsobuje určitú napäťovú stratu z 12 V napájania. Meraním sa zistilo že maximálne napätie, ktoré sme schopní na elektromagnete dosiahnuť je približne 10 V. Táto hodnota bude ďalej uvažovaná ako maximum dosiahnuteľného napäťia na elektromagnete (presná maximálna hodnota je zobrazená na Obr. 3.4, ale nakoľko môže jemne variovať v závislosti od napájacieho zdroja, prípadne iných faktorov, bola zaokrúhlená na 10 V).

4.1 Arduino IDE

Kedže počas vývoja MagnetoShield od R1 po R4 došlo k rozsiahlym zmenám, v tejto sekcií ponúkam podrobnejší náhľad do fungovania a popisu jednotlivých funkcií, ktoré sú súčasťou knižnice v jazyku C/C++. Celú knižnicu pozostávajúcu z hlavičky – angl. *header* a zdrojového kódu – angl. *source*¹, nájdete na stránke projektu [19], rovnako ako aj knižnice zabezpečujúce fungovanie ostatných shieldov.

Na začiatok by som v krátkosti priblížil *header* časť knižnice. Tento súbor má za úlohu importovať ďalšie potrebné knižnice, deklarovať globálne premenné, určiť ich dátový typ, či budú mať funkcie návratovú hodnotu, alebo vytvoriť konštanty pre knižnicu. Následne opíšem z používateľského hľadiska najdôležitejšie funkcie a keď bude potrebné, aj ich obsah definovaný v rámci časti *source*.

Ako si možno všimnúť, *header* sa skladá z dvoch častí. Prvá je napísaná pre preprocesor a príkazy sa začínajú prefixom `#`. Táto časť kódu sa vykonáva ešte pred kompliaciou kódu knižnice. Vhodnými príkazmi sme takto schopní do knižnice pridať ďalšie súbory, ktoré budú

¹ Ďalej v texte budem uvádzať anglické pomenovanie týchto súborov, nakoľko sú využívané bežnejšie.

súčasťou komplujúceho sa kódu, alebo určiť, ktoré časti knižnice sa majú zobrať do úvahy a ktoré nie.

Takýmto spôsobom sme napríklad pridali do knižnice dve ďalšie knižnice a to „*AutomationShield.h*“ a „*Wire.h*“.

```
#include "AutomationShield.h"
#include "Wire.h"
```

„*AutomationShield.h*“ v sebe obsahuje príkazy a prvky spoločné pre všetky zariadenia vytvorené v rámci iniciatívy AutomationShield a „*Wire.h*“ nám výrazne zjednoduší komunikáciu s DAC prevodníkom, ktorá prebieha pomocou I²C protokolu.

Na určovanie, ktoré časti knižnice sa majú brať do úvahy a ktoré nie, slúži zápis:

```
#ifndef SHIELDRELEASE
#define SHIELDRELEASE 4
#endif
```

Ide o to, že táto knižnica je primárne určená pre MagnetoShield R4, ale nakoľko R4 je len novou verziou vo vývojovom reťazci, je užitočné ak najnovšia verzia softvéru je kompatibilná aj so staršími modelmi. Práve tento príkaz určuje, s akou verzou MagnetoShield sa bude pracovať a tým pádom sa funkcie a konštanty fungujúce len so starým hardvérom neskompilujú. Túto selekciu si možno všimnúť v ďalších častiach knižnice, či už v *header* alebo *source* súbore. Napr. rozlíšenie, s akým typom DAC prevodníku sa bude komunikovať a nastavenie jeho adresy na základe MagnetoShield verzie:

```
#if SHIELDRELEASE==1 || SHIELDRELEASE==2
#define PCF8591 (0x90 >> 1)
#elif SHIELDRELEASE==3 || SHIELDRELEASE==4
#define MCP4725 (0x60)
#endif
```

Takže v prvej časti *header* sa definujú hlavné konštanty. Jedná sa o čísla predovšetkým opisujúce fyzikálne vlastnosti sústavy MagnetoShield R4. Tieto hodnoty a ich pôvod budú priblížené pri opise konkrétnej funkcie, v ktorej budú použité.

V ďalšej časti sú deklarované funkcie, teda sa určí, či bude mať funkcia návratovú hodnotu a ak áno, o aký dátový typ pôjde. Dátové typy zohrávajú pomerne veľkú úlohu, nakoľko zariadenia vedia pracovať len s dátami v podobe bitov (ako napr. DAC prevodník očakáva 12-bitovú správu, resp. 8-bitovú), ale pre používateľa je prirodzené zadávať číselné hodnoty v podobe reálnych čísel. Z tohto dôvodu treba byť veľmi opatrný, či sa jedná o funkciu posielajúcu dátu zo zariadenia alebo do neho.

Teraz sa posnažím priblížiť účel jednotlivých funkcií a ich previazanie s reálnymi meraniami. Celé formulácie kódu uvádzam nebudem, iba ak to bude nevyhnutné.

Prvou funkciou, ktorá inicializuje celé rozhranie vytvorené pre MagnetoShield je

```
void MagnetoShield.begin()
```

Jej účelom je „naštartovať“ knižnicu „*Wire.h*“ na sprostredkovanie I²C, určiť analógovú referenciu pre Arduino dosku a nastaviť rozlíšenie analógovo-digitálneho prevodníka v Arduine. Zariadenia Arduino UNO a MEGA fungujú prednastavene na 5 V TTL logike a majú vnútorný 10-bitový analógovo-digitálny prevodník (ďalej ADC prevodník – je rozdielny od už spomínaného DAC digitálno-analogového prevodníka). To znamená, že ak pošlem na analógový vstup do Arduina signál v rozmedzí 0 až 5 V, ADC prevodník ho premení na 10-bitové číslo a túto informáciu pošle mikroradiču. Táto zmena signálu sa označuje ako kvantizácia. Spojitý interval vstupných hodnôt sa následne skladá z 2¹⁰ častí, pričom každá z nich má veľkosť 5/1024 V, čo predstavuje limitné rozlíšenie s akým vieme čítať analógové dátu. Čo sa ale stane ak túto hodnotu napäťia –

referenciu zmeníme na 3.3 V? Samozrejme, prvé čo nám napadne je, že iba zariadenie obmedzíme v čítaní väčšieho signálu, čo sa zdá nežiadúce. Táto úvaha je sice pravdivá, ale takmer všetky vstupné signály idú cez napäťovú deličku, ktorej vieme vhodným odporom ľahko upraviť pomer a tie čo nejdú, majú maximálne výstupné napätie 2.5 V (Hallov senzor). Navyše pre 3.3 V referenciu platí to isté ako pre 5 V. Vstupný signál sa rozdelí na 2^{10} (1024) častí, ale v tomto prípade, už najmenší dielik má $3.3/1024$ V. Takže sme schopní zvýšiť presnosť s akou mikroradič číta analógový signál. Pre ešte vyššiu presnosť, dosky Arduino s mikroradičmi s ARM architektúrou (napr. Arduino DUE) majú zabudovaný 12-bitový ADC prevodník², takže signál sa v takomto prípade delí až na 2^{12} (4096) dielikov.

Kľúčovým prvkom knižnice je funkcia umožňujúca komunikáciu s DAC prevodníkom, keďže skrz tento komponent má používateľ dosah na akčný člen. Ako bolo už niekoľko krát naznačené, DAC prevodník prijíma správu od nadriadeného zariadenia (v tomto prípade Arduina) pomocou protokolu I²C, ako niekoľko bitovú správu. Rovnako ako u ADC prevodníka aj pri DAC platí, že maximálna veľkosť tejto správy určuje, aké rozlíšenie prevodník má. V tomto prípade meníme prijatú správu na interval 0 až 5 V. Nakoľko sa počas vývoja MagnetoShield DAC prevodník menil z 8-bitového na 12-bitový, aj základná funkcia obsluhujúca túto perifériu má 2 formulácie:

```
void MagnetoShield.dacWrite(uint8_t DAClevel)
void MagnetoShield.dacWrite(uint16_t DAClevel)
```

Rozdiel je v podstate len v argumente funkcie. V prvom prípade je do DAC prevodníka zasielaná 8-bitová správa, v druhom 16-bitová³. Výsledkom je v oboch prípadoch nastaviteľné napätie na výstupe prevodníka, ktoré má priamy vplyv na hodnotu napäťia na elektromagnete.

Spomenutá funkcia funguje na relatívne nízkej úrovni a pre používateľa je neprirodzené uvažovať o napäti, ako o bitovej správe. Preto bola vytvorená funkcia, ktorá rozumie vstupným hodnotám napäťia v reálnom tvaru. Používateľ zadáva hodnoty požadovaného napäťia na elektromagnete, teda hodnoty v intervale 0 až 10 V.

```
void MagnetoShield.actuatorWriteVoltage(float u)
```

Opäť aj tu máme 2 formulácie, jednu pre MagnetoShield R1 a R2 a druhú pre R3 a R4. Ich rozdielnosť je v tom, že R1 a R2 používajú MOSFET ako výkonový prvok veľkého a malého napäťia. Tým pádom, funkcia musí zabezpečiť mapovanie len tých DAC levelov, ktoré spôsobujú zmenu v napätí. Alternatívnejou formou volania tohto príkazu je aj `void MagnetoShield.actuatorWrite(float u)`.

So spomínaným maximálnym napäťím 10 V súvisí aj globálna premenná "voltageRef", ktorá v sebe nesie informáciu, aká je najvyššia možná dosiahnutelná hodnota napäťia na elektromagnete. Jej veľkosť je vopred nastavená na hodnotu 10 V. Je ju možné zobraziť pomocou príkazu `float MagnetoShield.getVoltageRef()` a zmeniť `void MagnetoShield.changeVoltageRef(float v_ref)`.

Podfunkciou, ak ju tak možno nazvať, ktorá priamo zodpovedá za vyriešenie závislosti napäti v prevode z DAC levelov na volty je

```
uint8_t MagnetoShield.voltageToDac(float vOut)
uint16_t MagnetoShield.voltageToDac(float vOut)
```

²Predvolene funguje s rozlíšením 10-bitov, kvôli kompatibilite s inými doskami. 12-bitové rozlíšenie je potrebné softvérovo zapnúť. V rámci mnou zostavenej knižnice pre MagnetoShield sa táto operácia deje automaticky.

³Maximálna dĺžka správy je 12-bitov, ale dátové typy v jazyku C alebo C++ štandardne nepokrývajú takúto veľkosť správy. Preto je správa formulovaná ako 16-bitová (2 bajty) a prvé 4 čísla sa vnútornie "orežú".

Nami žiadané napätie na elektromagnete je prepočítané na DAC levery, ktoré sú potrebné na jeho dosiahnutie. V prípade MagnetoShield R4 funkcia vracia 16-bitovú správu s maximálnou hodnotou 4095 (maximálna hodnota 12-bitového čísla). Táto hodnota je vypočítaná cez polynomiálnu funkciu

$$U_{DAC} = p_1 U_{out}^3 + p_2 U_{out}^2 + p_3 U_{out} + p_4, \quad (4.1)$$

kde U_{DAC} predstavuje výsledné DAC levery pre žiadané napätie definované ako U_{out} . Pri MagnetoShield R1 a R2 funkcia vracia 8-bitovú správu s maximálnou hodnotou 255, nakoľko je použitý 8-bitový DAC prevodník. Rozdiel oproti formulácii pre MagnetoShield R4 je aj v tom, že hodnota správy je prepočítavaná pomocou funkcie [20]

$$U_{DAC} = p_1 U_{out}^{p_2} + p_3 e^{U_{out} p_4}. \quad (4.2)$$

Ako som už uviedol, rozdielnosť výpočtu medzi vydaniami MagnetoShieldu spočíva v použití MOSFETu ako výkonového prvku pre vydania R1 a R2, zatiaľ čo pre R3 a R4 bol použitý opamp s NPN tranzistorom (viď Kap. 3). Konštanty funkcií p_1, p_2, p_3, p_4 boli aproximované pomocou rozhrania *MATLAB Curve Fitting Toolbox* a v knižnici sú definované v *header* súbore ako $P1, P2, P3, P4$.

Ako sa pri implementácii PID riadenia na MagnetoShield R1 ukázalo [15], reprezentácia vstupných a výstupných hodnôt v podobe % z celkového rozsahu vie byť veľmi užitočná.

```
void MagnetoShield.actuatorWritePercents(float u)
```

Táto funkcia spracováva zadaný vstup v % a podľa neho vytvorí odpovedajúcu hodnotu napäťia na elektromagnete.

Pri levitácii sa magnet vznáša voľne vo vzduchu. Napriek tomu má však určité ohraničenia priestoru, v ktorom takýto dej môže nastať. Nakoľko nemeríme jeho pozíciu priamo, ale pomocou Hallovho senzoru, potrebujeme vedieť, aký výstup posiela Hallov senzor v prípade, že sa magnet nachádza v hraničných polohách. Podľa týchto hodnôt a fyzikálnych parametrov ohraničenia je následne možné vytvoriť závislosť medzi výstupom zo senzora a fyzikálnou veličinou opisujúcou polohu magnetu. Funkcia, ktorá zisťuje výstup v hraničných polohách je

```
void MagnetoShield.calibration()
```

Výstupy zo senzora sú merané v dvoch pozíciah. Najprv sa magnet nachádza na dolnom ohraničení, na zemi. Veľkosť výstupu sa uloží do globálnej premennej “*minCalibrated*”. Pre presnosť merania a zaistenie, že naozaj bude zistený výstup zo senzora pre stav, kedy je magnet na zemi, sa meranie zopakuje 100 krát. Následne sa na DAC prevodník pošle správa na vytvorenie maximálneho napäťia, to vytiahne magnet do hornej polohy, kde narazí na horné ohraničenie. Opäťovne sa zmeria výstup zo senzora 100 krát a výsledná hodnota napäťia sa uloží do premennej “*maxCalibrated*”. Ak chceme zistiť hodnoty týchto premenných použijú sa funkcie *float MagnetoShield.getMinCalibrated()* a *float MagnetoShield.getMaxCalibrated()*. Súčasťou funkcie *MagnetoShield.calibration()* je aj, na základe nameraných hodnôt, vypočítať konštanty *d_p1*, *d_p2*. Tie sú využité pri definícii vzťahu medzi nameranou intenzitou magnetického poľa Hallovým senzorom a vzájomnou vzdialenosťou magnetu a elektromagnetu.

Zo spomenutého vyplýva, že táto funkcia je dôležitou súčasťou každého riadiaceho programu, nakoľko jej cieľom je presne špecifikovať hranice sústavy. V prípade, že sa táto funkcia z programu vypustí, v *header* sú definované predpočítané hodnoty pre všetky kalibratívne premenné.

Pre meranie pozície levitujúceho magnetu máme k dispozícii tri základné interpretácie. Prvá berie nameraný signál z Hallovho senzora a prepočítava ho na intenzitu magnetického poľa v Gaussoch.

```
void MagnetoShield.sensorReadGauss()
```

Koeficientom prevodu medzi nameraným výstupným napäťom zo senzora a Gaussami je $1.3 \text{ mV/G} = 769.23 \text{ G/V}$ a v knižnici je reprezentovaný konštantou *HALL_SENSITIVITY*. Tento prevod zabezpečuje funkcia `float MagnetoShield.adcToGauss(uint16_t adc)`.

Akonáhle máme k dispozícii informáciu o intenzite magnetického poľa a zistené hranice v ktorých môže magnet lietať, dá sa nameraný výstup z Hallovho senzora interpretovať v podobe %.

```
void MagnetoShield.sensorReadPercents()
```

Ak je magnet na hornej hranici funkcia vracia hodnotu 100 % ak je na dolnej, dostávame 0 %.

Posledným spôsobom ako interpretovať zmeranú indukcii magnetického poľa je vytvoriť funkciu vzájomnej pozície magnetov $h(B(t))$. Výsledkom tohto procesu je, že dostávame vzťah na určenie vzájomnej vzdialenosťi elektromagnetu a permanentného magnetu $h(t)$ na základe meniacej sa magnetickej indukcie $B(t)$. Funkcia, ktorá zabezpečuje túto konverziu je `float MagnetoShield.gaussToDistance(float g)`. V rámci nej je vzdialenosť prepočítaná pomocou vzťahu:

$$h(B(t)) = k_1 B(t)^{k_2} \quad (4.3)$$

Tvar funkcie opisujúcej vzťah medzi týmito fyzikálnymi veličinami bol prvotne odhadnutý a následne boli konštandy k_1, k_2 identifikované pomocou *MATLAB Curve Fitting Toolbox*. Konštandy k_1 a k_2 zodpovedajú spomenutým konštantám *d_p1, d_p2* z funkcie `MagnetoShield.calibration()`. Tieto konštandy sa určujú pri procese kalibrácie na základe nameraného signálu z Hallovho senzora. Vzťah na výpočet jednotlivých konštant je uvedený v kóde knižnice v časti *source*. V prípade neprebehnutia kalibrácie sa konštantám priradí prednastavená hodnota. Funkcia knižnice vracajúca hodnotu pozície v mm je

```
float MagnetoShield.sensorReadDistance()
```

Skrátené volanie tejto funkcie je `float MagnetoShield.sensorRead()`.

Zatiaľ som opísal funkcie monitorujúce a pracujúce s polohou magnetu v priestore a funkcie, ktoré ovplyvňujú a riadia akčný člen. Ale na kvalitné riadenie systému by bolo dobré poznať viacero fyzikálnych veličín charakterizujúcich deje v systéme. Jednou z týchto veličín je napätie na elektromagnete. Doteraz sme hovorili iba o tom, ako ho ovplyvniť a aká je jeho očakávaná hodnota. Ale zatiaľ nemáme funkciu sprístupňujúcu reálne namerané hodnoty. Funkcia umožňujúca zmerať napätie na elektromagnete je

```
void MagnetoShield.auxReadVoltage()
```

Hardvérová logika za týmto meraním bola opísaná v Kap. 3. Tu možno len spomenúť význam konštánt *VGAIN* a *ARES3V3*. Funkcia ako prvé zmeria hodnoty ADC levelov (pozor nie DAC, tu ide analógový signál do mikroradiča) na analógovom pine a následne ich násobí pomocou $ARES3V3 = \frac{3.3}{1024}$ alebo $ARES3V3 = \frac{3.3}{4096}$ konštant v závislosti od použitej dosky (1024 pre UNO/MEGA a 4096 pre ADC prevodník na DUE). Výsledok násobenia je hodnota napäťia, ktorú pin sníma. Na to, aby sme dostali hodnotu napäťia na elektromagnete, musíme ešte výsledok prenásobiť pomerom opačným ako je prenos napäťovej deličky, teda $VGAIN = 4$ (Prenos deličky určíme ako $G_d = \frac{R_2}{R_1+R_2}$, kde $R1 = 3 \text{ k}\Omega$, $R2 = 1 \text{ k}\Omega$).

Druhou fyzikálnou veličinou ktorej hodnotu chceme vedieť je elektrický prúd tečúci cez elektromagnet a zistíme ho zavolaním funkcie

```
void MagnetoShield.auxReadCurrent()
```

Prepočet snímaných ADC levelov na elektrický prúd je veľmi podobný tomu pri meraní napäťia. Tu ale vystupuje konštantu $IGAIN = 33.3333 \frac{mA}{V}$. Ako si možno všimnúť, konšstanta už necharakterizuje len bezrozmerný pomer výstupu ku vstupu, ale definuje závislosť medzi dvomi rozdielnymi fyzikálnymi veličinami a teda má aj prislúchajúcu jednotku.

Poslednou funkciou knižnice určenej pre rozhranie ArduinoIDE na obsluhu zariadenia MagnetoShield je

```
float MagnetoShield.referenceRead()
```

Úlohou tejto funkcie je snímať napätie na pine ku ktorému je pripojené napájanie 3.3 V a medzi nimi uložený potenciometer s nominálnym odporom $10\text{ k}\Omega$. Otáčaním potenciometra meníme jeho odpor a tým zvyšujeme alebo znížujeme napätie snímané pinom. Potom už závisí len na kreativite programátora, ako túto perifériu využiť. Jednou z možností je pomocou snímaného signálu interaktívne meniť napätie, ktoré posielala DAC prevodník a tým aj napätie na elektromagnete.

4.2 MATLAB a Simulink

Pre vytváranie riadiacich programov alebo simulácií vyšej zložitosti je vhodné používať programacie jazyky, ktoré už obsahujú určitý balík predprogramovaných funkcií. Takýmito jazykmi, alebo lepšie povedané rozhraniami, sú napríklad MATLAB a Simulink od *The Math Works*. Obe rozhrania sú v rámci iniciatívy AutomationShield využívané do veľkej miery a pre obe boli vyvinuté špecifické balíky funkcií, ktoré uľahčujú prístup používateľa k zariadeniam AutomationShield.

V rámci práce na MagnetoShield bol MATLAB využitý skôr na účely spracovania dát a identifikácie. Taktiež boli vytvorené simulácie riadenia na základe identifikovaného modelu⁴ a skripty inicializujúce nevyhnutné premenné a konštanty pre účely využitia v Simulinku. Okrem spomenutého, prostredníctvom MATLABu a softvéru *Multi-Parametric Toolbox 3* [9] bola tiež generovaná úloha explicitného prediktívneho riadenia. Využitím skriptov napísaných v MATLABe boli zostavené zdrojové kódy v jazyku Python a za pomoci softvéru *μAO-MPC* [31], boli generované algoritmy, ktoré mohli byť následne využívané, či už v Arduino IDE alebo v Simulinku pri použití prediktívneho riadenia (pre viac informácií a samotné skripty viď stránku projektu [19], sekcia „matlab/examples/MagnetoShield“).

Pre vytvorenie a nahrávanie riadiaceho programu ako takého, sa ale MATLAB ukázal ako nedostačujúci, napoko sa nejedná o jazyk fungujúci v reálnom čase - ide o interpretovaný jazyk. Z tohto dôvodu bol v rámci práce na MagnetoShield využívaný viac program Simulink. Ten poskytuje základnú možnosť pre vytváranie programov v reálnom čase a to tú, že program je možné vyexportovať do jazyku C a následne ho automaticky nahrať na hardvér, pričom súčasťou exportovania je určitá miera optimalizácie berúca do úvahy cielový hardvér. Preto bola pre Simulink vyvinutá aj špecifická knižnica, ktorú sa posnažím v nasledujúcich riadkoch priblížiť. Taktiež opíšem spôsob konfigurácie tohto rozhrania, aby vytvorené riadenie bolo možné na hardvér nahrať a zaznamenať namerané výsledky.

⁴skripty simulácií riadenia v rámci MATLABu neboli súčasťou mojej práce

Simulink je taktiež od *The MathWorks*, ale narozenie od MATLABu to je grafický jazyk, ktorý sa kompiluje. Využíva funkcie a skripty písané aj v jazyku, aký používa MATLAB, ale tie sú zabalené do špecifickej formy – funkcií nazývaných bloky. Tieto bloky okrem spracovávania MATLABových skriptov môžu poskytovať aj “obal” funkciám napísaných v iných jazykoch (napr. C,C++). Podstata grafickej reprezentácie kódu spočíva v tom, že uvedené bloky vykonáva jú proces na signále, ktorý je do bloku privedený. To umožňuje jednoduchšiu kontrolu programov, ktoré sa dajú vyjadriť ako sled cyklických za sebou idúcich a na seba nadväzujúcich operácií. To robí Simulink ideálnym prostredím pre vytváranie riadenia a simulácií dynamických systémov akým je napr. MagnetoShiel⁵. Aj prístup k funkciám je pomerne rozdielny. Okrem vstupných hodnôt signálu má väčšina blokov sadu parametrov, ktoré je možné konfigurovať podľa potreby po rozkliknutí daného bloku. Takže, nastavovanie argumentov funkcie prebieha v tomto prípade formou dialógového okna. Pre jednoduchú prácu s MagnetoShieldom boli dotvorené viaceré bloky zoskupené do rovnomennej podknižnice MagnetoShield. Spolu s ostatnými podknižnicami pre riadenie iných shieldov vytvárajú knižnicu AutomationShield. Pre správny chod knižnic v Simulinku ako aj využívanie funkcií vytvorených pre MATLAB je nutné nainštalovať “MATLAB Support Package for Arduino Hardware” a “Simulink Support Package for Arduino Hardware” od *The Math Works*.

Keďže sa jednotlivé typy prototypizačných dosiek Arduino od seba líšia (napr. rozlíšenie ADC prevodníka 10-bitov pre Arduino UNO, 12-bitov pre Arduino DUE), bolo potrené v rámci návrhu blokov vziať do úvahy aj túto skutočnosť. Preto podknižnica, ktorú na najbližších riadkoch opíšem, je navrhnutá tak, aby bola zaručene použiteľná pri práci s prototypizačnými doskami Arduino UNO, MEGA a DUE. Kompatibilita s inými zariadeniami nebola pri návrhu braná do úvahy. Opäť, pre viac informácií o knižnici, ako aj návod na jej inštaláciu a zdrojové kódy blokov, vid' stránku projektu [19].

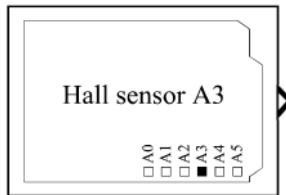
Prvým blokom podknižnice je blok *magnetoShieldPins* Obr. 4.1. Jedná sa o veľmi jednoduché sprostredkovanie nameraných dát na pine prototypizačnej dosky. Keďže MagnetoShield R4 využíva na prenos nameraných dát iba 4 piny, aj tento blok poskytuje používateľovi prístup iba k týmto 4 pinom⁶. Na ne sa prenášajú dáta z merania pozicie na Hallovej sonde, merania napäcia na elektromagnete, merania prúdu na elektromagnete a merania analógovej referencie z potenciometra. Tieto dáta sú zaznamenané ako 16-bitová správa, o rôznej maximálnej veľkosti v závislosti od použitej prototypizačnej dosky a ADC prevodníka v nej (UNO – 10 bitov maximálna dĺžka, DUE – 12 bitov maximálna dĺžka). Následne sa v rámci vnútornej štruktúry bloku konvertujú na dátový typ “double” a v takomto tvare sú aj používateľovi prístupné, takže používateľ dostáva na výstupe z bloku hodnotu v ADC leveloch, ale dátového typu double.

V rámci dialógového okna (Obr. 4.2) má používateľ, okrem možnosti zvolenia jednej z periférií (výber dosky Arduino), na ktorej sa bude meranie realizovať, aj možnosť výberu pinu z ktorého budú merané dáta zaznamenané. Taktiež je potrebné určiť vzorkovaciu periódus, teda ako často bude dané meranie prebiehať (pre odporúčané vzorkovania vid' Kap. 6, Tabuľka 6.1).

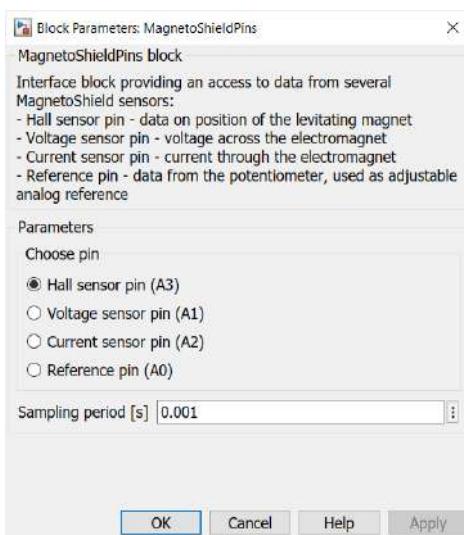
V závislosti od nastavenia pinu sa mení aj vzhľad obrázka bloku. To používateľovi uľahčuje orientáciu v programe.

⁵Proces identifikácie a analýza dát sú napr. procesy, ktoré je jednoduchšie robiť v skriptovacích jazykoch ako MATLAB. Vytváranie cyklickej štruktúry v Simulinku by bolo v tomto prípade neefektívne.

⁶MagnetoShield používa na komunikáciu s prototypizačnou doskou a na svoju prevádzku viacero pinov avšak signály, ktoré sa po nich prenášajú sú pre používateľa nepotrebné (napr. piny na komunikáciu I²C).



Obr. 4.1: Blok *magnetoShieldPins*

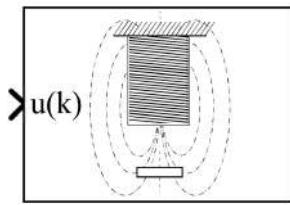


Obr. 4.2: Dialógové okno po otvorení bloku *magnetoShieldPins*.

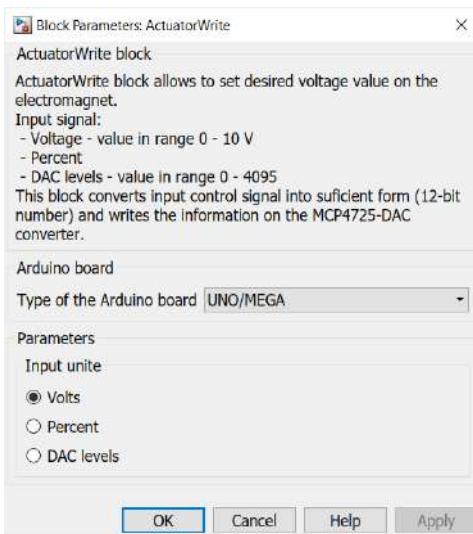
Rovnako, ako je potrebné vedieť čítať informácie o systéme, je dôležité mať možnosť signály do systému posielat. V prípade MagnetoShield R4 má systém jediný akčný člen – elektromagnet, ktorý sme schopní ovládať pomocou DAC prevodníku. Za zapisovanie informácií na DAC prevodník je zodpovedný blok *actuatorWrite* (Obr. 4.3). Vstupom do tohto bloku je opäť dátový typ double, ktorý je v rámci bloku konvertovaný na 16-bitovú správu s maximálnou hodnotou 4095 (DAC prevodník je totiž 12-bitový).

Po otvorení bloku *actuatorWrite* má používateľ v rámci kontextového okna k dispozícii dve nastaviteľné premenné. Ako prvé je potrebné nastaviť, aká doska Arduino je použitá pri riadení a komunikácii s MagnetoShield R4. Na výber sú dve možnosti a to Arduino UNO resp. MEGA alebo Arduino DUE (Obr. 4.4). Správa a jej štruktúra posielaná DAC prevodníku nie je závislá na použitej prototypizačnej doske. Dôvod, prečo je toto nastavenie potrebné vykonať je, že doska Arduino DUE má 2 komunikačné zbernice I²C, pričom UNO má len jednu. Zvolením typu dosky, na ktorom sa bude program vykonávať, sa nastaví správna zbernice pre vybranú dosku (UNO – 0, DUE – 1) a taktiež sa nastaví adresa DAC prevodníka, pomocou ktorej je Arduino schopné toto zariadenie osloviť a komunikovať s ním.

Funkcia vnútorne prepočítava vstup na DAC levels, preto je nutné vedieť maximálne hodnoty vstupu pre dané fyzikálne jednotky. Druhá nastaviteľná premenná uvádza o aké vstupné jednotky ide a zároveň definuje tomu odpovedajúce maximálne hodnoty.



Obr. 4.3: Blok *actuatorWrite*.



Obr. 4.4: Dialógové okno po otvorení bloku *actuatorWrite*.

S pomocou dvoch opísaných blokov, sme už schopní vytvárať programy pre MagnetoShield R4 a máme dosah ku všetkým jeho perifériam. Napriek tomu, podknížnica pre MagnetoShield obsahuje viacero blokov uľahčujúcich "písanie" programu. To pomáha používateľovi jednoducho implementovať logiku riadenia do svojho programu, bez potreby znalosti detailov ohľadom zariadenia.

Prvým takýmto blokom je *outputConverter* (Obr. 4.5). Jeho funkciou je prevádzdať výstupný signál zo systému na jednu zo zvolených fyzikálnych veličín. Tento blok teda vytvára rozhranie medzi nameraným signálom na pine a fyzikálnou jednotkou, ktorú si používateľ zvolí.

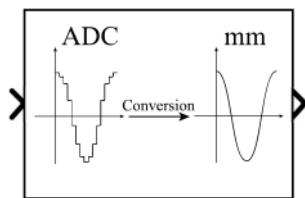
Po otvorení tohto bloku sa nám ukáže dialógové okno s niekoľkými možnosťami a nastaviteľnými parametrami (Obr. 4.6). V prvom rade je potrebné zvoliť prototypizačnú dosku. Blok *outputConverter* na základe zvolenej prototypizačnej dosky, zvolí rozlíšenie nameraného signálu a následné výpočty sa tomu prispôsobia. Na výstupe dostávame hodnotu dátového typu double vo zvolenej fyzikálnej jednotke:

- Position – Táto možnosť sa zvolí v prípade, že na vstup do bloku pripojíme meranie signálu z *magnetoShieldPins* a zvoleným pinom je "Hall sensor pin (A3)". V takomto prípade blok vie, že vstupujúce dátu sú pre výpočet polohy levitujúceho magnetu a na výstupe dostávame polohu v jednej z možných jednotiek – "Distance" (vzájomná vzdialenosť elektromagnetu a magnetu v milimetroch), "Percents" (tá istá vzdialenosť ale v percentách), "Gauss" (intenzita magnetického poľa magnetov), "ADC levels" (signál je prenesený na výstup v rovnakom tvare v akom vstupoval do bloku). Po vybratí jednotky "Distance" alebo "Percents" máme možnosť upraviť niekoľko konštánt potrebných pre vnútorný výpočet. Jedná sa o konštanty opísané v Kap. 4.1 pri popise funkcie *MagnetoShield.calibration()*

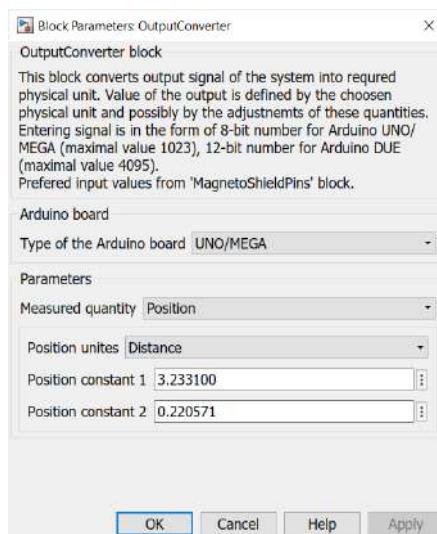
a `MagnetoShield.gaussToDistance()`. Tie určujú maximálne hranice pri prepočte signálu na percentá a udávajú tvar krivky pri prepočte na milimetre (Rov. (4.3)).

- Voltage – Táto možnosť je vhodná v prípade, že na vstup do bloku privádzame signál z pinu, na ktorom sa meria napätie, ktoré sa nachádza na elektromagnete. Zakliknutím “Advanced options” máme k dispozícii prestavenie konštant zabezpečujúcich prepočet nameraného signálu na napätie a prúd (analógia ku konštantám *VGAIN* a *IGAIN* na str. 23-24).
- Current – Pre správny výpočet elektrického prúdu na elektromagnete z dát meraných na pine, je vhodné zvoliť toto nastavenie. V “Advanced options” máme k dispozícii tie isté možnosti ako pri predošлом výbere.
- Analog reference – V prípade merania na “Reference pin (A0)”, táto možnosť mapuje nameraný signál na percentá a túto informáciu nám poskytuje na výstupe, pričom najmenší odpor predstavuje 0 % a najväčší 100 %.

Na základe vybraných nastavení sa mení aj grafika bloku. To má napomáhať používateľovi s ľahkou orientáciou vo vytvorenom programe.



Obr. 4.5: Blok *outputConverter*.



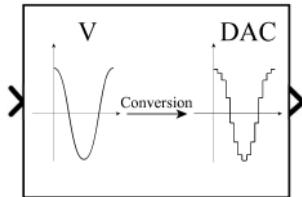
Obr. 4.6: Dialógové okno po otvorení bloku *outputConverter*.

Rovnako ako predchádzajúci blok menil signál, ktorému rozumie Arduino, na signál, ktorému rozumieme my, *inputConverter* blok (Obr. 4.7) robí presný opak. Nami vypočítané hodnoty sa na

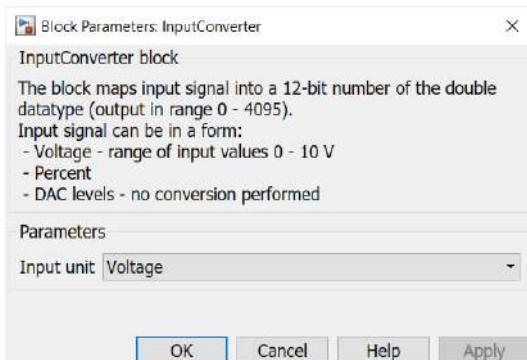
základe fyzikálneho kontextu transformujú na signál, ktorý je zapisovateľný na DAC prevodník⁷. Opäť máme k dispozícii niekoľko možností (Obr. 4.8), aký fyzikálny signál sa bude transformovať na DAC levels. Princípialne sa jedná o tie isté funkcie pre zapisovanie napäťia na DAC prevodník, ktoré boli opísané vyššie v Kap. 4.1.

- Voltage – Analógia k funkcií `void MagnetoShield.actuatorWrite(float u)`.
- Percents – Analógia k funkcií `void MagnetoShield.actuatorWritePercents(float u)`.
- DAC levels – Analógia k funkcií `void MagnetoShield.dacWrite(float u)`.

Funkcionalita tohto bloku už je zahrnutá v *actuatorWrite*, tým pádom nie je nutné aby sa tento blok nachádzal v riadiacom reťazci. Tento blok len rozširuje možnosti programátora pri riadení alebo simulácii. Opäťovne je využitá meniaca sa grafika pre jednoduchú čitateľnosť programu.



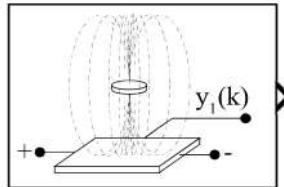
Obr. 4.7: Blok *inputConverter*.



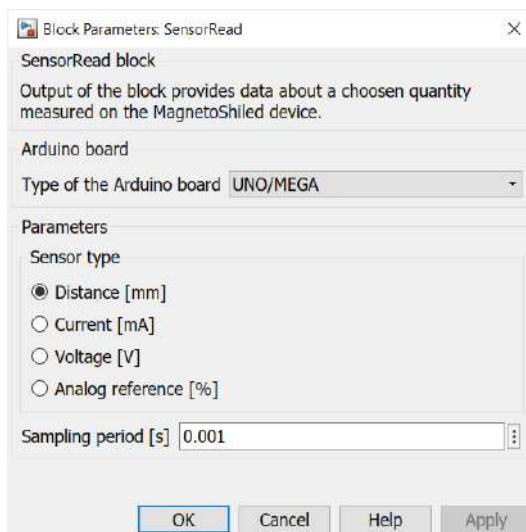
Obr. 4.8: Dialógové okno po otvorení bloku *inputConverter*.

Kombinácia blokov *magnetoShieldPins* a *outputConverter* poskytuje používateľovi plnú kontrolu nad meraniami v systéme MagnetoShield. Avšak vo väčšine riadiacich programov sú použité určité štandardné kombinácie týchto blokov a nastavenia ich parametrov. Preto kombináciou blokov *magnetoShieldPins* a *outputConverter* vznikol blok *sensorRead*. Táto funkcia vykonáva tie isté operácie na signály ako spomenuté bloky, avšak je okresaná o niekoľko nastaviteľných argumentov. Napomáha tým rýchlejšej tvorbe programov a eliminuje prípadne zdroje chýb.

⁷DAC prevodník očakáva 16-bitovú správu príslušného dátového typu. Tento blok vykonáva len transformáciu hodnôt, pričom dátový typ sa nemení, ostáva ako double.



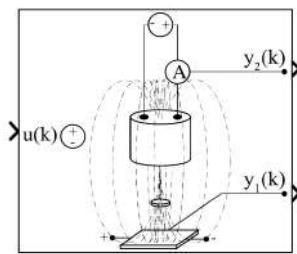
Obr. 4.9: Blok *sensorRead*.



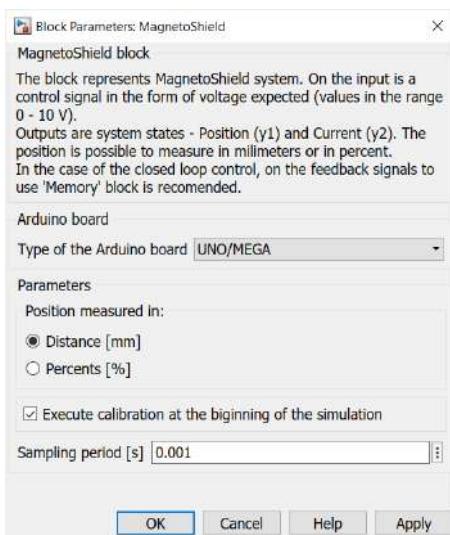
Obr. 4.10: Dialógové okno po otvorení bloku *sensorRead*.

Posledný blok podknižnice MagnetoShield je funkcia *MagnetoShield* reprezentujúca celý systém pomerne zjednodušeným spôsobom (Obr. 4.11). Má jeden vstupný port pre posielanie signálu do systému (napájacie napätie pre elektromagnet) a dva výstupné porty na meranie signálov stavových veličín. Je to z dôvodu, že úplná reprezentácia systému MagnetoShield v stavovom priestore vyžaduje okrem poznania polohy aj prúd tečúci cez elektromagnet (viac v Kap. 5).

Po otvorení dialógového okna máme k dispozícii opäť nastavenia dosky Arduino, spôsob reprezentácie nameranej polohy levitujúceho magnetu, vzorkovaciu periódu a možnosť vykonania kalibrácie. Zaškrtnutím tejto možnosti sa vykonajú rovnaké výpočtové úkony, aké su opísané v Kap. 4.1, funkcia *MagnetoShield.calibration()*. Tento proces sa vykonáva vždy na začiatku programu, ale výsledky sú prístupné len riadiacej jednotke a aj to len počas behu programu. Po ukončení, dátá získané kalibráciou zanikajú. Z tohto dôvodu odporúčam za účelom získania kalibračných výsledkov použitie programu *Calibration*, ktorý je súčasťou exemplárnych príkladov pre knižnicu MagnetoShield v prostredí Simulink (viď príklady pre MagnetoShield v Simulinku na stránke projektu [19], časť “*simulink/examples/MagnetoShield*”).



Obr. 4.11: Blok *MagnetoShield*.



Obr. 4.12: Dialógové okno po otvorení bloku *MagnetoShield*.

4.2.1 Konfigurácia Simulinku pre Arduino DUE

Dôležitou súčasťou vytvárania programu v Simulinku je konfigurácia pre použitý hardvér. Nakoľko je Simulink predovšetkým simulačné rozhranie, prekladanie vytvoreného programu na formu, ktorej Arduino porozumie je potrebné nastaviť. V rámci nastavení používateľ upresňuje, na aký hardvér sa program bude nahrávať, či bude povolená komunikácia medzi Simulinkom a hardvérom počas behu programu, prípadne iné nastavenia upresňujúce, akú formu má generovaný kód mať, aby ho cieľové zariadenie vedelo vykonávať. Nakoľko systém MagnetoShield vyžaduje pomerne vysokú frekvenciu vzorkovania, najideálnejším hardvérom z uvažovaných prototypizačných dosiek je Arduino DUE. Nižšie uvedený postup konfigurácie je uvažovaný predovšetkým pre tento hardvér.

Po otvorení nového modelu sa na hornej lište nachádza tlačidlo "Model Configuration Parameters".

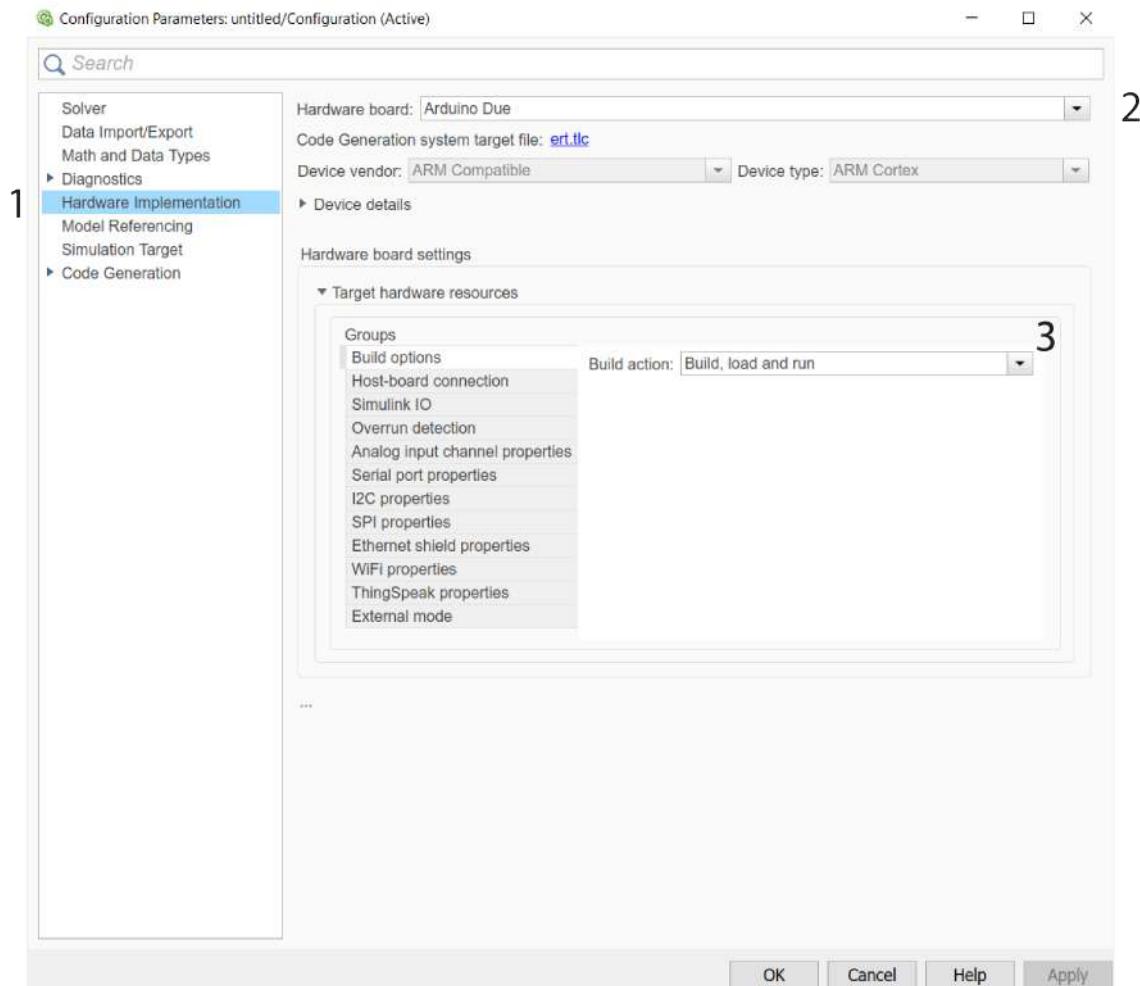


Nasledujúce číslovanie sa týka dialógového okna na Obr. 4.13.

- 1) V novootvorenom okne vyberieme záložku hardvérovej implementácie "Hardware Implementation" nachádzajúcu sa v ľavom menu.

- 2) Prvou nastaviteľnou možnosťou je voľba prototypizačnej dosky "Hardware board", kde vyberieme možnosť Arduino Due. Po ukončení tejto voľby a krátkej inicializácií sa v rámci "Hardware board settings" zobrazí možnosť otvorenia rozbaľovacieho menu "Target hardware resources".
- 3) Po otvorení tohto menu máme prístup k niekoľkým nastaveniam súvisiacim s tvorbou a implementáciou zdrojového kódu:
 - Prvé zo skupiny nastaviteľných parametrov je "Build option", ktoré zodpovedá za to, či sa kód programu má len zostaviť (vytvorenie kódu v jazyku C), alebo či sa má aj nahrať na cieľový hardvér a vykonať. Pre nahrávanie programu priamo do Arduina vyberieme *Build, load and run*.
 - "Host-board connection" definuje na ktorom USB porte je pripojené zariadenie. Manuálna voľba je potrebná v prípade, ak máme naraz zapojených viacero rovnakých prototypizačných dosiek. Pri prvotnej inštalácii hardvéru a jeho prvom použití, je možný výskyt problémov s rozoznávaním hardvéru. V tomto prípade je nutné uviesť číslo portu manuálne.
 - Možnosť "Simulink IO" dovoľuje používateľovi zasahovať do programu a meniť nastavenia niektorých blokov počas behu programu. Táto možnosť je veľmi nápomocná pri overovaní funkčnosti komunikácie a správnosti vytvoreného programu. Avšak podstatne spomaľuje chod celého programu a pri rýchлом systéme, akým je MagnetoShield, úspešné riadenie systému s touto možnosťou nie je možné.
 - Veľmi užitočnou pomôckou pri stavbe programov pre zariadenia s pomerne obmedzenou výpočtovou silou je "Overrun detection". Potvrdením tejto možnosti, používateľ určí jeden z pinov dosky Arduino, na ktorý bude poslaná log. 1 (v zmysle TTL logiky) v prípade, že sa zistí "pretekanie" hlavného programu. To znamená, že vzorkovanie je príliš rýchle a program nestihne vykonať všetky výpočtové operácie počas jednej períody. Následne je možné zostaviť jednoduchý elektrický obvod napájaný týmto pinom, spínajúci LED diódu. Tá nám toto pretekanie bude signalizovať.
 - Nastavenie "Analog input channel properties" určuje doske, čo je jej napäťovou referenciou pre vstupné analógové signály. Ako som už spomínał v Kap. 4.1, dosky UNO a MEGA majú prednastavenú 5 V logiku pre čítanie analógových signálov, pričom je možné túto hodnotu upraviť. Aby sme pre tieto dosky dostali lepšie rozlíšenie a využívali 3.3 V logiku, musíme nastavenie zmeniť z *Default* na *External*. To spôsobí, že systém za referenciu považuje signál, ktorý je na pine *ref*, kde privádzame práve 3.3 V (prepojenie pinu a zdroja je realizované prostredníctvom spojov na shielde). Pre Arduino DUE to v prípade Simulinku nie je nutné a 3.3 V referencia je predvolená.
 - "Serial port properties" slúži na nastavenie sériového prenosu medzi Arduinom a PC v jednotke baud.
 - Komunikácia medzi DAC prevodníkom a Arduinom prebieha cez protokol I²C. Hodnota je uvádzaná v Hz a dá sa v prípade potreby zmeniť (v príkladoch ktoré budú pre Simulink uvedené, bola použitá hodnota 400 000 Hz pre Arduino DUE – zrýchlenie chodu programu sa ale neprekázalo).
 - Nakoľko sa v MagnetoShielde, zariadenie využívajúce komunikačný protokol SPI nenačádza, táto možnosť sa spolu s ďalšími nastaveniami nevyužívaných periférií ponechá na pôvodných hodnotách.
 - Nastavenie "External mode" definuje, akým spôsobom prebieha komunikácia medzi PC a Arduinom počas behu programu. Vo všetkých vytvorených príkladoch bolo použité

nastavenie *Serial* (využívanie *XCP on Serial* je možné len v prípade dosiek s ARM architektúrou, napr. DUE).



Obr. 4.13: Postup konfigurácie dosky Arduino pri používaní MagnetoShieldu v prostredí Simulink.

Simulink poskytuje viacero spôsobov, ako prepojiť hardvér s PC a vytvárať alebo monitorovať programy bežiace či už na jednom alebo druhom zariadení. Spomenutý postup je uvedený hlavne z dôvodu, že toto prepojenie bolo využívané počas experimentov a je základom k úspešnému využívaniu exemplárnych príkladov, ktoré budú spomenuté v Kap. 6. Týmto spôsobom sme schopní priamo počas vykonávania experimentu zaznamenávať dátá v rozhraní Simulink, čo je výhodné pre jednoduchosť použitia, i keď za cenu straty niektorých údajov (bližšie opísané v závere Kap. 6.5).

V prípade využívania zaznamenaných dát pre účely identifikácie a analýzy systému odporúčam použiť iné rozhranie ako Simulink. Prípadným riešením je aj použitie blokov sprostredkujúcich sériovú komunikáciu, ktoré sú súčasťou toolboxu *Arduino Support from Simulink*. Následne je možné vyexportovať programu na Arduino bez toho, aby bolo udržiavané nepretržité prepojenie so Simulinkom (možnosť “*Deploy to Hardware*” – analógia k nahratiu programu cez Arduino IDE). Pomocou týchto blokov je možné neposielať dátá do Simulinku, ale iba ich vysielanie s určitou rýchlosťou na USB port Arduina. Potom je možné dátá čítať pomocou sériového terminálu počítaču napr. v Arduino IDE alebo aj v Simulinku spustením samostatného simulačného

programu čítajúceho sériové dátia pomocou toolboxu *Instrument Control Toolbox*. Tento postup ale funguje iba pre obmedzenú dĺžku dát, nakoľko maximálna prenosová rýchlosť nastaviteľná Simulinkom je 115 200 baud – byty za sekundu. Ak vezmeme do úvahy, že každé číslo posielané v rámci komunikácie má 4 alebo 8 bajtov (v závislosti od toho, či používame predvolený dátový typ double alebo vykonávame konverziu na single), prenos každého bajtu sériovou komunikáciou vyžaduje približne 10 bitov (start, stop a 8 bitová správa) a prenášame informáciu minimálne o 4 veličinách (3 stavy + vstup), dostávame 160 - 320 bitov potrebných na zaznamenanie jednej vzorky. Pri spomenutej prenosovej rýchlosťi je minimálny čas prenosu jednej správy 1.4 až 2.8 ms s tým, že tento výpočet uvažuje iba prenos správy. Treba vziať do úvahy aj čas na meranie veličín a chod programu, prijatie a spracovanie správy a prípadné komplikácie združujúce prenos. Z toho je zrejmá potreba vyššej prenosovej rýchlosťi alebo kódovania prenášaných údajov od špecifickej podoby s menšími nárokmi na veľkosť informácie⁸. Dosiahnutie vyššej rýchlosťi je možné priamy zásahom do automaticky generovaného kódu Simulinkom. Táto operácia je ale podstatne náročnejšia, ako implementácia riadenia pomocou iného aplikačného rozhrania a z tohto dôvodu sa tomu v rámci tejto práce venovať nebudem. To isté platí pre vytváranie špecifického kódovania pre prenos správ určených výhradne pre MagnetoShield.

4.3 Python

V rámci sprístupnenia práce s MagnetoShieldom širšiemu okoliu bolo vyvinuté rozhranie pre programovací jazyk Python. Python je typ interpretovaného programovacieho jazyka a vďaka tomu, skripty nie je nutné kompilovať. Na druhej strane je pre aplikáciu potrebný vyšší výkon zariadenia schopný interpretácie skriptu. Kvôli tomuto aspektu a kompatibilite je toto rozhranie navrhnuté a testované pre prototypizačné dosky od firmy Adafruit (Adafruit Metro M4 Express, Adafruit Metro M0 Express a Adafruit Metro M4 Grand Central) využívajúce jazyk CircuitPython, ktorý tvorí rozhranie medzi hardvérom dosky a Pythonom.

Okrem Python knižnice boli v rámci iniciatívy AutomationShield vytvorené aj exemplárne príklady. Tie sú principiálne rovnaké ako príklady vytvorené v jazyku C/C++ alebo Simulinku. Jedná sa o riadenie levitácie PID regulátorom, uložením pólov, LQ regulátorom a explicitným prediktívnym riadením (viac o jednotlivých typoch riadenia v Kap. 6). Nakoľko ich vytvorenie spoľu s Python knižnicou nebolo súčasťou mojej práce, v rámci ktorej som sa orientoval na prácu s doskami Arduino, detailnejšie sa im venovať nebudem (pre bližšie informácie vid' stránku projektu [19], sekcia “CircuitPython”).

Skripty napísané v Pythone sa okrem knižnice určenej pre prácu s MagnetoShieldom v tomto jazyku, využívajú aj na automatické generovanie častí knižníc a formuláciu úloh prediktívneho riadenia. Za týmto účelom bol prevzatý softvér na riešenie úloh prediktívneho riadenia pre lineárne systémy a generovanie príslušného algoritmu v jazyku C, *μAO-MPC: microcontroller Applications Online Model Predictive Control* [31]. Tento softvér akceptuje špecificky formulovanú úlohu kvadratickej optimalizácie definovanú pomocou skriptu, vytvoreného v Pythone. Tento skript slúži ako vstupné dátia a jeho úpravou je možné meniť aj zadávanú úlohu. Bližšie sa problematike prediktívneho riadenia a aj postupu ako narábať s *μAO-MPC* softvériom budem venovať v Kap. 6.7.

⁸Nakoľko vieme s akou presnosťou sú dátá merané a aká je ich maximálna/minimálna veľkosť, dala by sa informácia o nich uschovať do celočíselných dátových typov, ktoré sú menšie.

5 Modelovanie a identifikácia

Na prvotný návrh riadenia systému PID regulátorm, je nutné vedieť koľko má systém vstupov, výstupov, akú má približne dynamiku (aké vzorkovanie je potrebné na jeho riadenie) a či existuje nejaké ohraničenie na veľkosť vstupnej veličiny. Všetky tieto informácie o systéme dokážeme získať pomerne jednoducho. V tom spočíva výhoda PID regulácie, pomocou ktorej sme schopní dosiahnuť riadenie procesu bez potreby znalosti podrobnych informácií o systéme (samozrejme znalosť systému môže návrh PID riadenia výrazne zefektívniť a zlepšiť jeho správanie).

PID regulátor má okrem veľkých výhod aj určité nevýhody. Skutočnosť, že riadiaci vstup je vytváraný na základe regulačnej odchýlky (rozdiel žiadanej a nameranej hodnoty riadenej veličiny) spôsobuje, že regulátor do istej miery zanedbáva ostatné veličiny nachádzajúce sa v systéme a nezohľadňuje všetky procesy, ktoré môžu viac, či menej súvisieť s hodnotou riadenej veličiny.

Pri hlbšej analýze systému sa dá zistiť, že väčšinou neovplyvňujeme akčný člen priamo, ale prostredníctvom dynamického procesu, ktorý nie je na prvý pohľad zrejmý. Napríklad proces ovládania elektromagnetu – zmena DAC levelu mení elektrický prúd a až ten následne silu medzi magnetmi. To isté platí aj pri meraní fyzikálnych veličín. Ako bolo spomenuté v Kap. 4.1, pri meraní polohy levitujúceho magnetu v mm meríame v skutočnosti intenzitu magnetického poľa $B(t)$ a na základe prevodového vzťahu (4.3) vieme vypočítať pozíciu $h(B(t))$.

Ak dokážeme tieto deje nájsť a popísať ich, otvárajú sa nové možnosti náhľadu na systém a jeho chovanie. Okrem riadenia jedinej veličiny vieme zrazu odhadnúť oveľa väčšie množstvo dejov a na základe toho s nimi efektívnejšie pracovať a kontrolovať ich.

Za účelom opisu systému, jeho vstupov, výstupov a jeho najdôležitejších veličín (stavov) budeme používať vyjadrenie v podobe diferenciálnych rovníc prepísaných do špecifickej formy zvanej stavová reprezentácia. Na základe tohto modelu sme následne schopní podrobne analyzovať systém, predikovať jeho budúce chovanie a navrhovať riadiace algoritmy (riadenie uložením pólov, LQR, MPC,...).

Mimo spomenutých výhod má stavová reprezentácia osobitý význam pri návrhu a výrobe nového zariadenia, nakoľko porovnaním očakávaných hodnôt stavov so skutočnými, nameranými alebo odhadnutými hodnotami sme schopní odhaliť a prípadne aj bližšie určiť hardvérovú, softvérovú alebo aj modelovú chybu.

Spomenuté vlastnosti boli počas vytvárania mojej diplomovej práce dostačujúcou motiváciou pre nájdenie matematickej interpretácie MagnetoShieldu, ktorá by do čo najväčzej miery systém vystihovala.

Už v rámci práce na MagnetoShield R2 bol vytvorený článok *“MagnetoShield: Prototype of a Low-Cost Magnetic Levitation Device for Control Education”* [20]. V rámci neho bol uvedený model opisujúci sily a deje pri magnetickej levitácii realizovanej elektromagnetom. Súčasťou návrhu tohto modelu bol aj proces identifikácie. Tá prebiehala na vzorke dát získaných pomocou riadenia využívajúceho PID regulátor. Celkovým výsledkom identifikácie bol pomerne presný model, využiteľný na účely rôznych modelovo orientovaných algoritmov riadenia. Tento model

bol aj hlavným modelom pri práci na exemplárnych príkladoch reprezentujúcich používanie jednotlivých rozhraní pre MagnetoShield v rámci *open-source* iniciatívy AutomationShield. Pre viac informácií o tomto modeli a príkladoch demonštrujúcich jeho použitie, viď Kap. 6¹ a stránku projektu [19], sekcia “*simulink/examples/MagnetoShield*”.

Tento model využíval predpoklad, že levitujúcim telesom je feromagnetická gulička, sila je vyjadrená na základe koenergie poľa $W' = \frac{1}{2}L(x)I^2$ a funkcia indukčnosti $L(x)$ je závislá od vzájomnej vzdialenosťi guličky a elektromagnetu [11]. Nakoľko ale MagnetoShield nevyužíva len silu z elektromagnetu, ale zásadnú a hlavnú úlohu hrá pole generované neodymovým magnetom, považoval som za potrebné navrhnúť model, ktorý by túto neopomenuteľnú zložku sily skúsil zohľadniť. Preto som sa rozhodol navrhnúť vlastný model tohto systému, ktorý sa pokúsím v najbližších podkapitolách priblížiť.

5.1 Základný koncept

Úlohou elektromagnetického levitátora, akým je aj MagnetoShield, je dosiahnuť ustálenú polohu permanentného magnetu vo voľnom priestore, na základe žiadanej výšky levitácie. Ak sa na túto úlohu pozrieme z fyzikálneho hľadiska, dostaneme na základe Newtonových pohybových zákonov rovniciu

$$\vec{F} = \vec{F}_g + \vec{F}_{\text{mag}}, \quad (5.1)$$

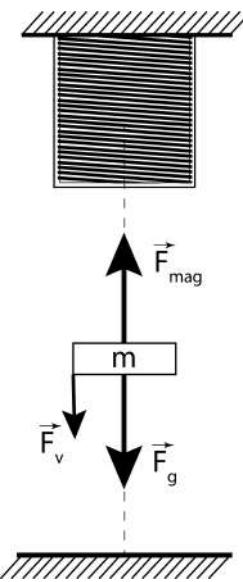
teda výsledná sila \vec{F} pôsobiaca na permanentný magnet vo vzduchu je rovná rozdielu tiažovej sily \vec{F}_g a sily magnetickej \vec{F}_{mag} , ktorá vzniká medzi magnetom a elektromagnetom.

Vytvorenie tejto rovnice je veľmi priamočiare a vychádza z vlastností systému, ktoré sú na prvý pohľad veľmi ľahko badateľné a v princípe sa vyskytujú v každom dynamickom systéme, podliehajúcemu účinkom gravitácie.

Ale ak sa nad rovnicou a Newtonovými zákonmi hlbšie zamyslíme, zákon zotrváčnosti uvádza, že hmotný bod ostáva v pokoji alebo rovnomerne priamočiarom pohybe, ak suma všetkých naň pôsobiacich síl je nulová. Takže táto rovnica nie je ešte celkom tým fyzikálnym modelom, ktorý hľadáme, lebo nezaručuje, že magnet sa nebude hýbať z požadovanej výšky levitácie. Ak aj vyvážim magnetickú силu \vec{F}_{mag} s tiažovou silou \vec{F}_g , magnet môže byť v pokoji, ale aj v pohybe. Do Rov. (5.1) preto treba pridať ešte jednu silu eliminujúcu aj možnosť pohybu magnetu. Tým, že je táto sila závislá od pohybu, v prípade že je magnet v pokoji, zaniká a možno ju definovať ako tlmenie sústavy. Túto novú silu označím ako \vec{F}_v . Smer všetkých uvažovaných síl pôsobiacich na permanentný magnet s hmotnosťou m je zobrazený na Obr. 5.1 (orientácia a veľkosť sily \vec{F}_v závisí od smeru pohybu magnetu). Pohybová rovnica pre tento “konceptný” model vo vektorovom tvare je

$$\vec{F} = \vec{F}_g + \vec{F}_{\text{mag}} + \vec{F}_v = \vec{0}. \quad (5.2)$$

¹Kap. 6 síce uvažuje využitie iného modelu, ktorého opisu sa budem v tejto časti venovať, ale uvedená teória je aplikovateľná na akýkoľvek platný model.



Obr. 5.1: Sily pôsobiace na levitujúci magnet.

5.2 Silové zložky pohybovej rovnice

Akonáhle máme definované základné sily pôsobiace na teleso a máme predpoklad ako vznikajú, môžeme Rov. (5.2) prepísať do skalárneho tvaru ako:

$$m \frac{d^2r(t)}{dt^2} = -mg + F_{\text{mag}}(t) - c \frac{dr(t)}{dt}, \quad (5.3)$$

kde $r(t)$ predstavuje veľkosť polohového vektora, určujúceho pozíciu magnetu, jeho prvá derivácia definuje rýchlosť magnetu a druhá derivácia jeho zrýchlenie. Konštantu g je tiažové zrýchlenie a c predstavuje konštantu vyjadrujúcu vplyv tlmenia na dynamiku systému. Z tohto dôvodu sa tu nachádza aj záporné znamienko. Otázkou ostáva, od akých veličín závisí magnetická sila F_{mag} , ktorou sa vzájomne magnet a elektromagnet prítahujú?

Prvá dôležitá vec, ktorú bolo potrebné si uvedomiť pri hľadaní vzťahu, ktorý vystihuje silové účinky dvoch magnetických polí je, že výsledné pole od magnetu a elektromagnetu má príliš komplikovanú štruktúru. Presný opis takéhoto poľa by bol časovo veľmi namáhavý a vzhľadom na cieľ použitia tohto modelu aj príliš komplexný.

Z tohto dôvodu bolo potrebné nájsť určitú abstrakciu modelu, ktorá by bola dostatočne jednoduchá, aby sa dala ďalej využívať, no popri tom vystihovala chovanie sa daného javu.

Jedna z možností, ako sa dá pozerat na magnet rôznorodého tvaru a štruktúry, je uvažovať o ňom ako o magnetickom dipóle. Takže hovoríme o určitom zjednodušení komplikovaného poľa tým, že zanedbáme vplyv tvaru magnetu a budeme o ňom uvažovať, ako o obyčajnom symetrickom magnete so severným a južným pólom tvaru gule. Rovnaké pole generuje aj vodivá prúdová slučka [25, str. 265]. Čím ďalej sa od tvarovo komplikovaného zdroja magnetického poľa nachádzame, tým je pole podobnejšie poľu dipólu. Ak polia od elektromagnetu a aj permanentného magnetu budeme uvažovať s takýmito vlastnosťami, značne to uľahčí modelovanie sín vznikajúcich medzi nimi. Taktiež, nakoľko sa magnety k sebe nikdy príliš nepriblížia, môžeme považovať model za pomerne presný, aj keď so zmenšujúcou sa vzdialenosťou jeho presnosť bude klesať, nakoľko vplyv tvaru magnetu bude zásadnejší.

Ako je známe, elektrický prúd pretekajúci vodičom vytvára rovnaký typ poľa, aké tvoria magnetické látky (Ampérov zákon). Tým pádom je možné vyjadriť vlastnosti magnetickej látky vytvorením elektrickej slučky vhodnej veľkosti, ktorou preteká primeraný prúd. Vznikajúce pole má rovnaký charakter, ako pole od vyššie spomenutého dipólu a teda obe takéto polia možno opísať veličinou magnetický moment dipólu \vec{m} , ktorý budeme pre rozlišenie od hmotnosti značiť s dolným indexom \vec{m}_i , kde i charakterizuje objekt, ku ktorému sa vzťahuje. Nahradením elektromagnetu dipólem sme značne zjednodušil závislosť magnetických sín od tvaru elektromagnetu. Tým, že poznáme jeho rozmery a prúd ktorý ním preteká, možno vypočítať magnetický moment ako [25, str. 264]:

$$\vec{m}_i = I \vec{S}, \quad (5.4)$$

kde I je elektrický prúd tečúci elektromagnetom a \vec{S} je vektor plochy elektromagnetu. V takomto prípade, ale plocha S nie je zodpovedajúca rozmerom elektromagnetu, nakoľko tento model vyjadruje magnetické vlastnosti prostredníctvom jediného závitu. Intuitívne, aby model hodnovernejšie vyjadroval vlastnosti elektromagnetu pomocou vektoru plochy \vec{S} , je nutné vzťah uvažovať pre každý závit zvlášť. Preto dostávame konečný vzťah pre magnetický moment dipólu², ktorým sme nahradili elektromagnet

$$\vec{m}_i = N I \vec{S}, \quad (5.5)$$

kde N predstavuje počet závitov elektromagnetu. Ak nie je toto číslo uvedené ako parameter elektromagnetu, je možné jeho približnú hodnotu odhadnúť na základe merania magnetickej indukcie \vec{B} pomocou Hallovho senzoru, alebo zmerať indukčnosť cievky L a na základe toho odhadnúť počet závitov.

Pri výpočte približného počtu závitov pomocou Hallovho senzora možno vychádzať zo vzťahu, udávajúceho veľkosť magnetickej indukcie B solenoidu, meranej na jeho osi vo vzdialosti x v skalárnom tvaru [25, str. 267]:

$$B = \frac{\mu NI}{2} \left[\frac{x}{\sqrt{R^2 + x^2}} + \frac{l-x}{\sqrt{R^2 + (l-x)^2}} \right]. \quad (5.6)$$

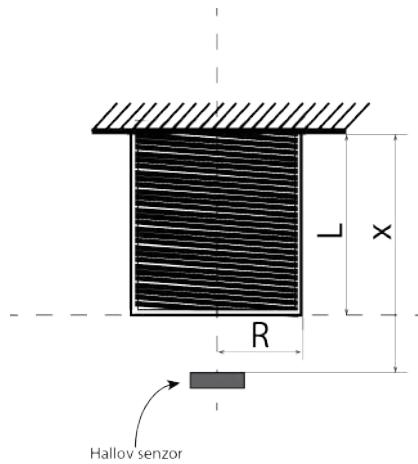
Konšstanta μ je permeabilita železného jadra nachádzajúceho sa v strede solenoidu a počíta sa ako $\mu = \mu_0 \mu_r$, kde μ_0 je permeabilita vakuu a μ_r je relatívna permeabilita jadra elektromagnetu, ktorá má hodnotu $\mu_r = 5000$ [26]³. I je konštantný elektrický prúd tečúci cievkou, R je polomer elektromagnetu a l predstavuje dĺžku elektromagnetu (l a R predstavujú rozmery navinutej cievky okolo železného jadra, bez vonkajšieho obalu). Ak sa meracie zariadenie dá do pozície $x = l$ (Obr. 5.2) výsledný vzťah pre veľkosť magnetickej indukcie a z toho odvodeneho počtu závitov na elektromagnete je:

$$B = \frac{\mu NI}{2} \frac{x}{\sqrt{R^2 + x^2}}, \quad (5.7)$$

$$N = \frac{2B\sqrt{R^2 + x^2}}{\mu x I}. \quad (5.8)$$

²Neberieme do úvahy priestorové rozloženie závitov.

³Materiál jadra elektromagnetu je neznáma zliatina železa, takže relatívna permeabilita je uvedená pre železo čistoty 99.8%, čo by mohol byť použitý materiál.



Obr. 5.2: Základné rozmery pri výpočte počtu závitov cievky.

Vykonaním merania dôjdeme k jednotlivým hodnotám veličín potrebných na výpočet závitov a k približnej hodnote počtu závitov v cievke. Výstupom z Hallovho senzora bolo napätie $U_1 = 2.45$ V, ak cievkou prúd I nepretekal a $U_2 = 1.951$ V, ak cievkou prúd I pretekal. Citolivosť sondy je $s = 1.3 * 10^{-3}$ V/G. Potom výsledná magnetická indukcia bola vypočítaná ako $B = \frac{U_1 - U_2}{s} * \frac{1}{10000} = 0.0384$ T. Treba tu však upozorniť, že výpočet je iba približný, nakoľko meranie bolo realizované za použitia nie ideálnej meracej techniky a nemáme informáciu o materiály z ktorého je stred elektromagnetu (ak má materiál horšie magnetické vlastnosti, ako tie uvedené pomocou relatívnej permeability, teda hodnota μ_r by bola menšia, výsledný počet závitov bude väčší).

Veličina	Hodnota
B – Zmeraná magnetická indukcia	0.0384 T
R – Polomer elektromagnetu	0.0062 m
x – Pozícia Hallovej sondy	0.0126 m
I – Zmeraný prúd	0.1021 A
μ – Permeabilita feritu	$6.3 * 10^{-3}$ H/m
N – Počet závitov	210

Tabuľka 5.1: Hodnoty veličín a parametrov pre výpočet počtu závitov N.

Nakoľko sa dá vyjadriť magnetické pole elektromagnetu pomocou magnetického momentu dipólu, je výhodné takýmto spôsobom opísť taktiež vlastnosti poľa od permanentného magnetu. V tomto prípade je ale pole tvorené na základe iného javu, ako je tok elektrického prúdu v slučke. Preto na popis \vec{m}_i použijeme veličinu nazývanú magnetizácia \vec{M} . Tá udáva hustotu magnetického dipólového momentu v prostredí definovanom objemom V , teda charakterizuje magnetické vlastnosti látky vytvárajúcej pole [2, 8]:

$$\vec{M} = \frac{d\vec{m}_i}{dV}, \quad (5.9)$$

$$\vec{m}_i = \vec{M}V = \frac{1}{\mu_0} \vec{B}_r V. \quad (5.10)$$

Konštanta B_r sa nazýva remanencia a určuje magnetické pole, ktoré si materiál zachová, ak na neho prestane pôsobiť vonkajšie magnetické pole. Toto číslo sa zvyčajne udáva ako súčasť

charakteristických vlastností materiálu a pre neodym N38, z ktorého je permanentný magnet vyrobený, má hodnotu (1,22 – 1,25) T (týka sa 8 mm magnetu, pre materiál 9 mm magnetu vidľ tabuľka 3.1) [5].

Dôvod, prečo je dobré mať oba zdroje magnetického poľa vyjadrené ako dipóly je ten, že vzťah pre vzájomnú silu medzi dvomi dipólmi je známy [29]:

$$\vec{F}_{\text{mag}}(t) = \frac{3\mu_0}{2\pi r(t)^4} \left[(\vec{m}_1 \cdot \hat{r})\vec{m}_2 + (\vec{m}_2 \cdot \hat{r})\vec{m}_1 + (\vec{m}_1 \cdot \vec{m}_2)\hat{r} - 5\hat{r}(\vec{m}_1 \cdot \vec{r})(\vec{m}_2 \cdot \vec{r}) \right]. \quad (5.11)$$

Vo výraze (5.11) predstavuje premenná \vec{m}_1 magnetický moment pre elektromagnet a \vec{m}_2 predstavuje magnetický moment permanentného magnetu, μ_0 je permeabilita vakuua (resp. prostredia medzi magnetom a elektromagnetom, ale to tiež nepoznáme), r je veľkosť polohového vektora určujúceho pozíciu jedného dipólu vzhľadom na druhý a \hat{r} je jednotkový vektor udávajúci smer pre polohu r . Po uvážení iba jedného rozmeru a to v osi z vzťah možno prepísat do skalárneho tvaru:

$$F_{\text{mag}}(t) = -\frac{3\mu_0 m_1 m_2}{2\pi r(t)^4}, \quad (5.12)$$

z čoho vyplýva, že smerovanie magnetickej sily bude opačné polohovému vektoru určujúcemu polohu levitujúceho magnetu.

Pred tým, ako vytvoríme už kompletnejšiu rovnici opisujúcu dynamiku medzi dvoma magnetmi, je potrebné si ešte ujasniť orientáciu jednotlivých vektorov a na základe toho aj znamienka vystupujúce v rovnici.

Ako prvé si treba určiť samotnú vzťažnú sústavu. Na základe princípu, akým funkcia `float sensorReadDistance()` meria pozíciu magnetu v priestore, opísaného v Kap. 4.1, polohový vektor \vec{r} určuje vzdialenosť levitujúceho magnetu od elektromagnetu (viď Obr. 5.3). Z tohto dôvodu sa počiatok súradného systému nachádza na povrchu elektromagnetu a aj orientácia vektora je určená jednotkovým vektorom \vec{r} smerom nadol. Vektor \vec{S} je vektor plochy a jeho orientácia je v smere severného pólu vznikajúceho pretekáním prúdu I cez elektromagnet⁴. Remanencia B_r opísaná vyššie má rovnaký smer ako má magnetický moment a ten je orientovaný v smere severného pólu magnetu. Posledná vektorová veličina zobrazená na obrázku je tiažové zrýchlenie \vec{g} , ktorého orientácia je smerom k zemi.

Na základe orientácie \vec{r} je vidno, aký vplyv majú jednotlivé sily na sústavu MagnetoShield R4 (Obr. 5.4). Veľmi dôležitým zistením je, že tiažová sila spôsobuje nárast veľkosti polohového vektora \vec{r} a magnetická sila ho naopak zmenšuje. Tomu je nutné prispôsobiť aj orientáciu súradného systému, ktorého kladné hodnoty rastú smerom k zemi.

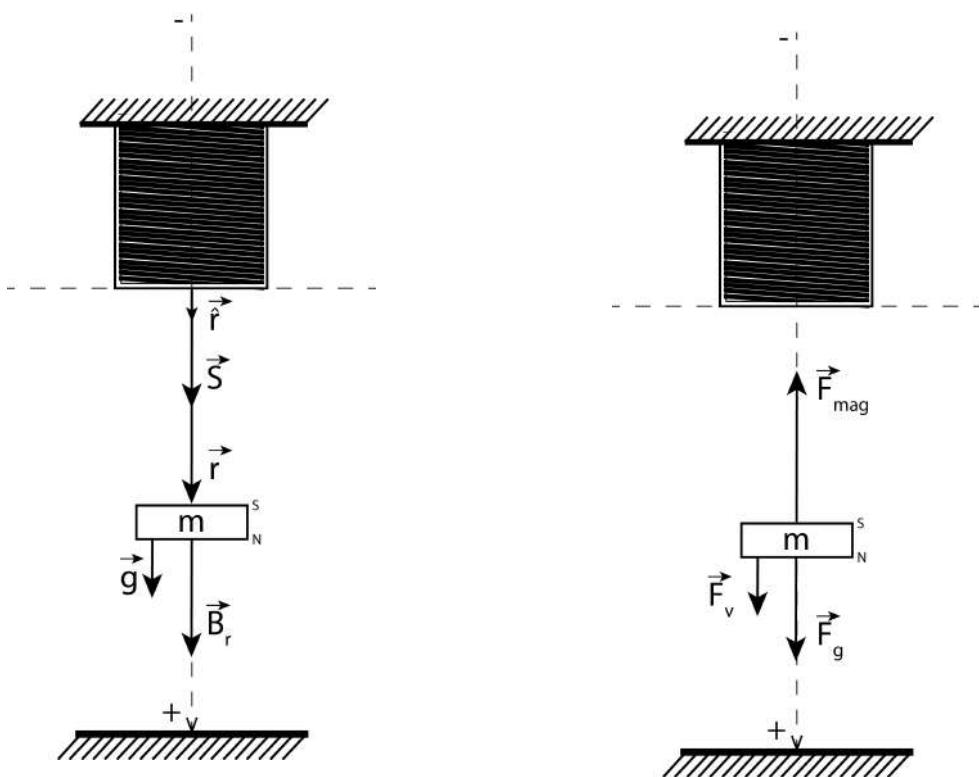
Po zohľadnení orientácie jednotlivých veličín, výsledný vzťah pre výpočet veľkosti magnetickej sily vznikajúcej medzi elektromagnetom a permanentným magnetom je:

$$F_{\text{mag}}(t) = -\frac{3SNB_rVI(t)}{2\pi r(t)^4} = -K \frac{I(t)}{r(t)^4}, \quad (5.13)$$

kde konšstanta K predstavuje hodnotu závisiacu na konštrukčných vlastnostiach oboch magnetov.

Na záver, na základe úpravy súradného systému zobrazenej na obrázkoch Obr. 5.3 a Obr. 5.4 a na základe Rov. (5.13), je nutné zmeniť znamienka v pohybovej rovnici 5.3. Vplyv zmeny

⁴Na ktorom konci elektromagnetu vzniká severný pól bolo určené na základe meraní Hallovým senzorom. Ten je schopný merať oba typy magnetických polí. Ak naň nepôsobí vonkajšie pole, má na výstupe napätie 2,5V v prípade detekcie poľa od severného pólu napätie klesá. Čím je permanentný magnet nižšie, tým menšie napätie meriame na výstupe Hallovho senzoru. Z toho sa dá usúdiť, že severný pól permanentného magnetu je orientovaný na dol a teda to isté musí platiť aj o elektromagnete.



Obr. 5.3: Súradnicový systém a vektorové veličiny. Obr. 5.4: Súradnicový systém a orientácia síl.

orientácie súradného systému na F_{mag} je už zahrnutý v odvádzaní rovnice Rov. (5.13), takže zmena znamienka prebieha iba pri tiažovej sile mg . Po osamostatnení druhej derivácie polohy dostávame vzťah opisujúci pohyb levitujúceho magnetu:

$$\frac{d^2r(t)}{dt^2} = g - \frac{KI(t)}{mr(t)^4} - c\frac{dr(t)}{dt}. \quad (5.14)$$

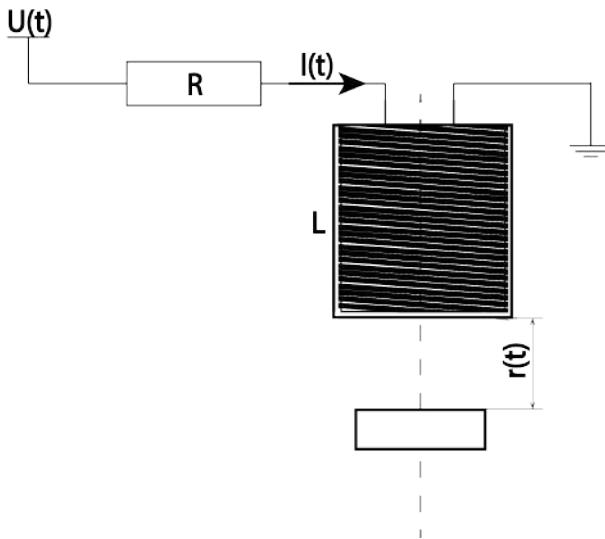
Hodnoty jednotlivých konštant a zmeraných veličín potrebných pre výpočet konštanty K , ako aj jej hodnota sú uvedené v tabuľke:

Veličina	Hodnota
S – Plocha elektromagnetu	0.1021 m ²
N – Počet závitov	210
Br – Remanencia neodymu	1.26 T
V – Objem permanentného magnetu	1.0053 * 10 ⁻⁷ m ³
K – Magnetická konšstanta systému	1.5307 * 10 ⁻⁹ N·A ⁻¹

Tabuľka 5.2: Veličiny a parametre pre výpočet magnetickej konštanty K.

5.3 Elektrická rovnica

V predošlej podkapitole bolo uvedené, akým spôsobom ovládaná magnetická sila pôsobí na levitujúci magnet a určili sa podmienky pre jav elektromagnetickej levitácie. Vychádzali sme z predpokladu, že sme schopní vytvárať magnetickú silu, definovanú Rov. (5.13), na základe elektrického prúdu pretekajúceho cievkou elektromagnetu. V Kap. 3 pri opise hardvérového rozhrania



Obr. 5.5: Zjednodušenie elektrického obvodu pre napájanie elektromagnetu.

a taktiež v Kap. 4 pri opise softvérového rozhrania sme uvažovali o ovládaní elektromagnetu pomocou napäťia, nie prúdu. Nakoľko sú ale tieto dve veličiny na sebe priamo závislé, nájdením vhodného vzťahu sme schopní vyjadriť ovládanie magnetickej sily $F_{mag}(t)$ pomocou meniaceho sa napäťia $U(t)$.

V Kap. 3 na Obr. 3.5 vidno celkovú schému elektrického obvodu zariadenia MagnetoShield R4. Pre nájdenie fyzikálnych vzťahov, definujúcich konverziu vytváraného napäťia pomocou DAC prevodníka na elektrický prúd, sa stačí zamerat' len na elektrický obvod zabezpečujúci napájanie elektromagnetu.

Operačný zosilňovač spoločne s tranzistorom slúžia na reguláciu napájacieho napäťia pre elektromagnet, možno teda o nich uvažovať, ako o nastaviteľnom zdroji napäťia $U(t)$. Veľkosť elektrického odporu **R8** spoločne s pasívnymi odpormi jednotlivých komponentov z Obr. 3.5 možno nahradiať jediným odporom R primeranej veľkosti. Cievka elektromagnetu je zadefinovaná pomocou veličiny vlastná indukčnosť L . V obvode sa ešte nachádza dióda, ale jej vplyv na obvod môžeme zanedbať nakoľko slúži len ako ochrana. Po uvážených zjednodušeniach je možné zobraziť čiastkový obvod charakterizujúci napäťové a prúdové deje pri riadení na Obr. 5.5.

Okrem elektrického obvodu je na Obr. 5.5 zobrazený aj permanentný magnet vo vzdialenosťi r od elektromagnetu. Je to z dôvodu vplyvu elektromagnetickej indukcie na elektrický obvod. Je dôležité mať na pamäti vzťah meniaceho sa elektrického a magnetického poľa. Nakoľko sa ale jedná o komplikovanú a rozsiahlu tému, uvediem len niekoľko kľúčových vzťahov, ktoré ma viedli uvažovať o elektrických pomeroch v obvode tak, ako budú uvedené.

Faradayov zákon elektromagnetickej indukcie hovorí: "Elektromotorické napätie indukované v každom závite cievky v akomkoľvek obvode je rovné zápornej časovej zmene magnetického toku v cievke". Teda ak mám magnetické pole vytvorené cievkou, ktoré sa práve rúti-zmenšuje sa, energia ktorá v ňom bola uschovaná v podobe magnetického poľa sa mení na elektromotorické napätie,

$$\varepsilon = -\frac{d\Phi(t)}{dt}. \quad (5.15)$$

Z uvedenej rovnice možno usúdiť, že v obvode napájajúcom elektromagnet odchádza k tomuto javu rovno dva krát (použité vzťahy [18, str. 593-602]).

V prvom prípade mám vytvorené magnetické pole na elektromagnete, pomocou ktorého sa snažím dosiahnuť jav levitácie. Tým pádom, nutne toto pole musí meniť svoju veľkosť v čase (musí sa prispôsobovať tak, aby udržalo magnet v žiadanej polohe) a na základe Rov. (5.15) vzniká na cievke napätie ε . Vďaka zápornému znamienku viem, že toto napätie bude opačné ku zmene poľa a ďalej v rovnici ho budem vnímať ako istý druh odporu⁵. Tento jav sa nazýva vlastná elektromagnetická indukcia.

Tu sa dá použiť ešte jedna úvaha. Ako som už spomenul, pretekajúci elektrický prúd vytvára magnetické pole, tým pádom meniacu hustotu magnetického poľa je možné vyjadriť aj v tendenciách meniacich sa elektrického prúdu [18, str. 601]:

$$\varepsilon = -L \frac{dI(t)}{dt}. \quad (5.16)$$

Konšanta L sa nazýva vlastná indukčnosť a je to charakteristický parameter pre danú cievku.

Druhý jav elektromagnetickej indukcie vzniká vďaka levitujúcemu permanentnému magnetu. Ak vezmem do úvahy, že jeho pole je konštantné, ale magnet sa pohybuje voči cievke, tiež možno hovoriť o meniacom sa magnetickom poli z pohľadu cievky. Rovnako na základe Rov. (5.15) dochádza ku generovaniu elektromotorického napäťia. Toto napätie, ale naopak od predošlého prípadu je zapríčinené vonkajšou prácou, takže oňom v ďalších rovniciach budeme uvažovať ako o zdroji napäťia.

Rovnako ako pri prvom prípade, aj tu je potrebné meniace sa magnetické pole vyjadriť v tendenciách známych a merateľných veličín. Hustota magnetického poľa $\Phi(t)$ hovorí o toku magnetickej indukcie $\vec{B}(t)$ cez uzavretú plochu \vec{S} , teda

$$\Phi(t) = \vec{B}(t) \cdot \vec{S}. \quad (5.17)$$

Po dosadení do Rov. (5.15) a uvážení, že práve vektor indukcie je veličina meniacia sa v čase, rovnica pre napätie ε generované v cievke pohybom magnetu má tvar

$$\varepsilon = -N \vec{S} \frac{d\vec{B}(t)}{dt}. \quad (5.18)$$

Konšanta N je v rovnici práve z toho istého dôvodu, ako tomu bolo v Rov. (5.5). Teda napätie sa naraz generuje vo veľkom počte závitov. Tu možno opäťovne využiť aproximáciu vlastností permanentného magnetu pomocou dipólu. Magnetická indukcia magnetického dipólu je v priestore opísaná rovnicou [25, str. 265]

$$\vec{B}(t) = \frac{\mu_0}{4\pi} \frac{3\hat{r}(\hat{r} \cdot \vec{m}_i) - \vec{m}_i}{r(t)^3} \quad (5.19)$$

a po uvážení len jedného rozmeru možno túto rovnicu zapísť v skalárnom tvare ako

$$B(t) = \frac{\mu_0 m_i}{2\pi r(t)^3}. \quad (5.20)$$

Premenná \vec{m}_i je magnetický moment dipólu nahradzajúceho permanentný magnet, $r(t)$ je vzdialenosť medzi elektromagnetom a permanentným magnetom a \hat{r} je smerový vektor v osi z . Ďalej ak zderivujeme vzťah (5.20) podľa času t , dostávame

$$\frac{dB}{dt} = \frac{d}{dt} \left(\frac{\mu_0 m_i}{2\pi r(t)^3} \right) = -\frac{3\mu_0 m_i}{2\pi r(t)^4} \frac{dr(t)}{dt}. \quad (5.21)$$

⁵ Ak napätie napájajúce elektromagnet vzrástie, zväčší sa aj pole, čo vyžaduje určitú energiu, takže ε sa bude chovať ako odpor. Naopak ak napájacie napätie klesne, pole sa začne rútiť a vytvorí napätie na cievke, akoby sa tam objavil nový zdroj napäťia.

Následne túto rovnicu spolu s rovnicou pre magnetický moment permanentného magnetu (5.10) dosadíme do vzťahu (5.18):

$$\varepsilon = \frac{3SNB_rV}{2\pi r(t)^4} = K \frac{1}{r(t)^4} \frac{dr(t)}{dt}. \quad (5.22)$$

Za povšimnutie stojí možnosť vyňať zo vzťahu parametre nezávislé od času. Výsledkom je rovnaká konštantă K , aká sa nachádza v Rov. (5.13). V ďalších krokoch uvediem, ako spomenuté javy ovplyvňujú elektrický obvod pri riadení levitácie na MagnetoShielde.

Na základe 2. Kirchhoffovho zákona možno pre obvod na Obr. 5.5 napísat rovnicu

$$U(t) - RI(t) - L \frac{dI(t)}{dt} + K \frac{1}{r(t)^4} \frac{dr(t)}{dt} = 0. \quad (5.23)$$

Prvý člen, ako bolo uvedené predstavuje nastaviteľné napájacie napätie. Druhý člen vyjadruje na základe Ohmovho zákona napäťovú stratu na súčiastkach s odporom a z toho dôvodu má aj záporné znamienko. Tretí a štvrtý člen zohľadňujú vplyv elektromagnetickej indukcie na obvod. $L \frac{dI(t)}{dt}$ má záporné znamienko na základe Rov. (5.16) z ktorej vyplýva, že napätie na cievke bude vždy opačné zmene elektrického prúdu v obvode. Posledný člen je kladný a možno o ňom uvažovať ako o externom meniacom sa zdroji napäťia. Polarita je daná na základe orientácie vzťažnej sústavy a vykonaného experimentu. Pri ňom sa potvrdilo, že priblženie magnetu (južným póлом) spôsobuje nárast magnetického poľa a napäcia nameraného na cievke. Výsledný elektrický model systému je potom daný rovnicou

$$U(t) = RI(t) + L \frac{dI(t)}{dt} - K \frac{1}{r(t)^4} \frac{dr(t)}{dt}. \quad (5.24)$$

Po jej miernej úprave dostaneme diferenciálnu rovnicu popisujúcu zmeny prúdu na elektromagnete v závislosti na napäti:

$$\frac{dI(t)}{dt} = -\frac{R}{L}I(t) + \frac{1}{L}U(t) + K \frac{1}{Lr(t)^4} \frac{dr(t)}{dt}. \quad (5.25)$$

5.4 Lineárny model

V predošlých podkapitolách som uviedol 2 rovnice, opisujúce javy v systéme MagnetoShield R4 z hľadiska dynamických a aj elektrických vlastností a to, ako som o nich uvažoval. Dohromady z nich dostávame nelineárny model systému:

$$\begin{aligned} \frac{d^2r(t)}{dt^2} &= g - K \frac{I(t)}{mr(t)^4} - c \frac{dr(t)}{dt}, \\ \frac{dI(t)}{dt} &= -\frac{R}{L}I(t) + \frac{1}{L}U(t) + K \frac{1}{Lr(t)^4} \frac{dr(t)}{dt}. \end{aligned} \quad (5.26)$$

Okrem premenných času t sa tu nachádza aj niekoľko konštánt. Ich hodnota vychádza buď z fyzikálnych vlastností použitého komponentu, alebo z teoretických úvah, ktoré boli uvedené vyššie, pričom tieto úvahy môžu viac alebo menej reprezentovať skutočnosť. Z tohto dôvodu bolo potrebné systém na základe vytvorennej predlohy identifikovať metódou tzv. "grey-box" identifikácie.

Tento proces hľadá hodnoty konštánt a parametrov systému opísaného rovnicami tak, aby vytvorený model čo najvernejšie reprezentoval namerané dáta. Samotné hľadanie mení iba hodnoty parametrov, štruktúra modelu sa ale nemení.

Rovnice v (5.26) sú nelineárne závislosti premenných $r(t)$, $\frac{dr(t)}{dt}$, $I(t)$. To výrazne komplikuje hľadanie správnych hodnôt parametrov. Taktiež MagnetoShield je systém, ktorý nie je len nelineárny, ale bez späťnej väzby nestabilný. Takže bez vhodného riadenia ani nie sme schopní namerať dátu, na základe ktorých by sme mohli systém identifikovať [20]. Práve kvôli týmto vlastnostiam systému sa v procese identifikácie vynára niekoľko netriviálnych problémov, ktoré nie je jednoduché prekonať. Síce existujú spôsoby, ako aj takýto nelineárny systém identifikovať, ale ich objasnenie nie je hlavným cieľom tejto práce.

Možnosť ako pomerne efektívne identifikovať správne parametre systému je zmeniť ho na lineárny systém. Nelineárny opis systému je sice z fyzikálneho hľadiska zmysluplnejší a aj presnejší, ale ako som už spomenul, jeho tvorba a práca s ním je veľmi komplikovaná a nepriamočiara. Lineárna reprezentácia má tiež tú výhodu, že na ňu možno aplikovať veľké množstvo poznatkov z teórie riadenia a lineárnej algebry.

Za týmto účelom je potrebné model opísaný v Rov. (5.26) linearizovať. Nakoľko v ďalších krokoch budeme využívať formu zápisu modelu v stavovom priestore, je jednoduchšie konverziu do tejto formy urobiť ešte pred linearizáciou.

Pre vytvorenie modelu v stavovom tvaru je nutné spraviť substitúciu vstupnej veličiny, napäťia $U(t)$ a stavových veličín $r(t)$, $\frac{dr(t)}{dt}$, $I(t)$, teda polohy, rýchlosť a prúdu, uvedeným spôsobom:

$$\begin{aligned} u(t) &= U(t), \\ x_1(t) &= r(t), \\ x_2(t) &= \frac{dr(t)}{dt} = \dot{x}_1(t), \\ x_3(t) &= I(t). \end{aligned} \tag{5.27}$$

Následne sa dajú Rov. (5.26) zapísat ako tri diferenciálne rovnice prvého rádu doplnené o rovnicu merania, ktorá definuje, aké stavy sme na systéme schopní merať. Výsledná stavová reprezentácia nelineárneho modelu má formu

$$\begin{aligned} \dot{x}_1(t) &= x_2(t), \\ \dot{x}_2(t) &= g - K \frac{x_3(t)}{mx_1(t)^4} - cx_2(t), \\ \dot{x}_3(t) &= -\frac{R}{L}x_3(t) + \frac{1}{L}u(t) + K \frac{1}{Lx_1(t)^4}x_2(t), \\ y &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}. \end{aligned} \tag{5.28}$$

Pri linearizácii sa využíva aproximácia pomocou rozvoja do Taylorovho radu, čo je polynomiálna reprezentácia funkcie v podobe nekonečného radu:

$$\begin{aligned} f(x) &= f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \dots \\ &\quad + \frac{f^n(x_0)}{n!}(x - x_0)^n + \dots \end{aligned} \tag{5.29}$$

V tejto rovnici premenná x_0 reprezentuje nejakú konkrétnu hodnotu premennej x . Aby sa získala lineárna funkcia, je potrebné vziať do úvahy iba prvé 2 členy tohto rozvoja (ostatné vyjadrujú už nelineárne závislosti) a teda,

$$f(x) = f(x_0) + f'(x_0)(x - x_0). \tag{5.30}$$

Je nutné povedať, že aproximácia pomocou prvých 2 členov zaručuje správne hodnoty pre model len v bode x_0 (tzv. linearizačný bod) a v jeho bezprostrednom okolí. Čím ďalej sa budeme nachádzať od bodu x_0 , tým bude väčšia šanca, že lineárna funkcia (5.30) bude odlišná od pôvodnej funkcie. Táto skutočnosť vyzdvihuje význam správneho zvolenia bodu x_0 , v ktorého okolí sa funkcia linearizuje.

Pre jednoduchosť zápisu na základe vzťahu (5.28) vykonajme nasledovné substitúcie:

$$\begin{aligned}\dot{x}_2(t) &= f_d(\vec{x}(t)), \\ \dot{x}_3(t) &= f_e(\vec{x}(t), u(t)).\end{aligned}\tag{5.31}$$

Premenné x a x_0 z Rov. (5.30) nahradíme vektormi \vec{x} a \vec{x}_0 , pričom \vec{x} je vektor stavových veličín $x_1(t), x_2(t), x_3(t)$ a $u(t)$ je vstupná veličina ($u(t)$ nepíšem ako vektor nakoľko vieme, že v tomto prípade je vstupom jediná veličina, napäťie). Vektor \vec{x}_0 bude reprezentovať vektor všetkých linearizačných hodnôt, v ktorých okolí sa model linearizuje, teda x_{01}, x_{02}, x_{03} a nakoľko je model závislý aj od vstupnej veličiny $u(t)$, linearizačný bod vstupu bude ďalej označený ako u_0 .

Po vložení $f_d(\vec{x}(t))$ a $f_e(\vec{x}(t), u(t))$ do Rov. (5.30) a uvážení, že je potrebné model derivovať parciálne podľa každej premennej, dostávame zo systému (5.28) systém lineárnych rovníc definovaný ako⁶

$$\begin{aligned}\dot{x}_1(t) &\approx x_{02}(t) + \frac{\partial x_2(t)}{\partial x_2} \Big|_{\vec{x}=\vec{x}_0} (x_2(t) - x_{02}), \\ \dot{x}_2(t) &\approx f_d(\vec{x}_0) + \frac{\partial f_d(\vec{x}(t))}{\partial x_1} \Big|_{\vec{x}=\vec{x}_0} (x_1(t) - x_{01}) + \frac{\partial f_d(\vec{x}(t))}{\partial x_2} \Big|_{\vec{x}=\vec{x}_0} (x_2(t) - x_{02}) \\ &\quad + \frac{\partial f_d(\vec{x}(t))}{\partial x_3} \Big|_{\vec{x}=\vec{x}_0} (x_3(t) - x_{03}), \\ \dot{x}_3(t) &\approx f_e(\vec{x}_0, u_0) + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial x_1} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} (x_1(t) - x_{01}) + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial x_2} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} (x_2(t) - x_{02}) \\ &\quad + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial x_3} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} (x_3(t) - x_{03}) + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial u} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} (u(t) - u_0).\end{aligned}\tag{5.32}$$

Prvým členom v rovniciach je hodnota funkcií v bodoch, v ktorých okolí linearizujeme, určených konštantami \vec{x}_0 resp. u_0 . Otázkou ostáva, aké majú byť hodnoty týchto konštánt?

Cieľom levitátora MagnetoShield je dostať magnet vo volnom priestore do momentu, kedy sa voľne vznáša – je v rovnováhe. Tento fakt napovedá, že model by mal opisovať chovanie sa systému práve v oblasti rovnováhy resp. v jej blízkosti. To znamená, že zmena jednotlivých stavov by mala byť nulová. Aby to platilo, musí taktiež platiť,

$$\begin{aligned}x_{02} &= 0, \\ f_d(x_0) &= g - K \frac{x_{03}}{mx_{01}^4} - cx_{02} = 0, \\ f_e(x_0, u_0) &= -\frac{R}{L}x_{03} + \frac{1}{L}u_0 + K \frac{1}{Lx_{01}^4}x_{02} = 0.\end{aligned}\tag{5.33}$$

Nakoľko sa jedná o sústavu troch rovníc so štyrmi premennými, jedna premenná ostáva volná. Ako je z vyššie uvedeného vzťahu vidno, na to aby bol systém v rovnováhe, rýchlosť x_{02} musí byť

⁶Rovnicu merania z modelu (5.28) v najbližších operáciach neuvažujeme. Táto rovnica je rovnaká ako pre nelineárny, tak aj lineárny model.

rovná 0. Síce máme tým pádom zadefinovanú jednu veličinu a ostávajú len tri neznáme, ale na ich výpočet máme k dispozícii už len dve rovnice. Takže stále musíme nejakú premennú zvoliť. Za voliteľnú veličinu je prirodzené vybrať polohu x_{01} , nakoľko to je aj používateľom nastaviteľná veličina pri procese riadenia.

Stále ale ostáva otázka, ako túto hodnotu x_{01} správne zvoliť? Prakticky môžeme funkciu linearizovať v okolí ktoréhokoľvek bodu rozsahu, v ktorom je možné levitáciu na MagnetoShielde vykonávať. Najefektívnejším riešením je zvoliť strednú hodnotu výšky levitácie $x_1(t)$ z dát, na základe ktorých budeme systém identifikovať. Akonáhle máme hodnotu x_{01} zvolenú, možno ju dosadiť do Rov. (5.33). Ale tento výpočet je založený na znalosti hodnôt konštánt, ktoré súce približne poznáme, ale ich hodnota sa môže v rámci procesu identifikácie meniť. Takže rovnice (5.33) sú pravdivé iba v prípade, že už prvotný odhad konštánt je dokonale presný, čo samozrejme nemusí platiť (to je aj dôvod prečo robíme identifikáciu). Tento problém je možné jednoducho vyriešiť na základe reálnych dát a to tak, že opäť za linearizačné body zvolíme stredné hodnoty prúdu $x_3(t)$ a napäcia $u(t)$. Rýchlosť $x_2(t)$ ostáva rovná 0, nakoľko to je podmienka ustálenej levitácie. Takže problém zvolenia vhodných linearizačných bodov sme zmenili na úlohu zvolenia vhodného experimentu, z ktorého budú dátá použité.

Na účely identifikácie som použil experiment s LQ riadením. Voľbu tohto riadenia za účelom zberu dát vysvetlím neskôr v Kap. 5.5. Hodnoty jednotlivých linearizačných bodov získaných na základe experimentu sú následovné:

$$\begin{aligned} x_{01} &= 0.0149 \text{ m}, \\ x_{02} &= 0 \text{ ms}^{-1}, \\ x_{03} &= 0.0236 \text{ A}, \\ u_0 &= 5.0056 \text{ V}. \end{aligned} \quad (5.34)$$

Mimo týchto operácií zavedieme ešte substitúciu, ktorou nahradíme derivácie stavových veličín a výrazy v zátvorkách $(\vec{x}(t) - \vec{x}_0)$ a to spôsobom,

$$\Delta\vec{x}(t) = \vec{x}(t) - \vec{x}_0, \quad (5.35)$$

$$\dot{\vec{x}}(t) = \Delta\vec{x}(t) + \dot{\vec{x}}_0. \quad (5.36)$$

Na základe úvahy z Rov. (5.33) je možné z Rov. (5.32) vyškrtnúť prvý člen nakoľko je rovný 0 a tak isto aj $\dot{\vec{x}}_0 = 0$ z Rov. 5.36. Následne sa systém (5.32) zmení na podobu:

$$\begin{aligned} \Delta\dot{x}_1(t) &\approx \frac{\partial x_2(t)}{\partial x_2} \Big|_{\vec{x}=\vec{x}_0} \Delta x_2(t), \\ \Delta\dot{x}_2(t) &\approx \frac{\partial f_d(\vec{x}(t))}{\partial x_1} \Big|_{\vec{x}=\vec{x}_0} \Delta x_1(t) + \frac{\partial f_d(\vec{x}(t))}{\partial x_2} \Big|_{\vec{x}=\vec{x}_0} \Delta x_2(t) + \frac{\partial f_d(\vec{x}(t))}{\partial x_3} \Big|_{\vec{x}=\vec{x}_0} \Delta x_3(t), \\ \Delta\dot{x}_3(t) &\approx \frac{\partial f_e(\vec{x}(t), u(t))}{\partial x_1} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} \Delta x_1(t) + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial x_2} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} \Delta x_2(t) \\ &\quad + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial x_3} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} \Delta x_3(t) + \frac{\partial f_e(\vec{x}(t), u(t))}{\partial u} \Big|_{\substack{\vec{x}=\vec{x}_0 \\ u=u_0}} \Delta u(t). \end{aligned} \quad (5.37)$$

Dôležitým faktom, ktorý treba vziať do úvahy je, že teraz systém nereprezentujú priamo stavové veličiny poloha $x_1(t)$, rýchlosť $x_2(t)$ a prúd $x_3(t)$ ale ich zmena oproti hodnote x_0 . To isté platí aj pre vstupnú veličinu $u(t)$. Tento princíp treba následne aplikovať aj pri práci s modelom⁷.

⁷Napríklad pri validácii modelu je nutné od vstupných a nameraných dát ako prvé odčítať hodnoty $x_{01}, x_{02}, x_{03}, u_0$, až potom je možné model s nameranými dátami porovnať.

Posledným krokom linearizácie je v Rov. 5.37 nahradieť členy $f_d(x)$ a $f_e(x, u)$ a vykonať parciálne derivácie. Výsledkom je linearizovaný model pre MagnetoShield R4:

$$\begin{aligned}\Delta \dot{x}_1(t) &= \Delta x_2(t), \\ \Delta \dot{x}_2(t) &= \frac{4Kx_{03}}{mx_{01}^5} \Delta x_1(t) - c\Delta x_2(t) - \frac{K}{mx_{01}^4} \Delta x_3(t), \\ \Delta \dot{x}_3(t) &= -\frac{4Kx_{02}}{Lx_{01}^5} \Delta x_1(t) + \frac{K}{Lx_{01}^4} \Delta x_2(t) - \frac{R}{L} \Delta x_3(t) + \frac{1}{L} \Delta u(t).\end{aligned}\quad (5.38)$$

Nakoľko sa jedná už o čisto lineárne vzťahy, do maticovej formy možno prepísať už aj rovnice definujúce dynamiku systému. Tento tvar je už známa reprezentácia lineárneho systému v stavovom priestore:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t).\end{aligned}\quad (5.39)$$

Člen $Du(t)$ nachádzajúci sa v rovnici merania v Rov. (5.39) uvádzam len na tomto mieste. V ďalšom teste ho nebudeme brať do úvahy.

Takže po uvážení maticovej formy a priradení hodnoty $x_{02} = 0$ výsledný model nadobúda tvar

$$\begin{aligned}\begin{bmatrix} \Delta \dot{x}_1(t) \\ \Delta \dot{x}_2(t) \\ \Delta \dot{x}_3(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ \frac{4Kx_{03}}{mx_{01}^5} & -c & -\frac{K}{mx_{01}^4} \\ 0 & \frac{K}{Lx_{01}^4} & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \\ \Delta x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \Delta u(t), \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \\ \Delta x_3(t) \end{bmatrix}.\end{aligned}\quad (5.40)$$

Ak vezmeme v úvahu, že elektrický odpor obvodu má približne hodnotu $R = 210 \Omega$, indukčnosť elektromagnetu bola určená na $L = 0.239 \text{ H}$ a na základe výpočtov a ich výsledkov v tabuľkách 5.1 a 5.2, vieme zostaviť prvotný, odhadovaný model MagnetoShieldu:

$$\begin{bmatrix} \Delta \dot{x}_1(t) \\ \Delta \dot{x}_2(t) \\ \Delta \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 263.2003 & 0 & -41.6924 \\ 0 & 0.1313 & -893.6170 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \\ \Delta x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 4.2553 \end{bmatrix} \Delta u(t). \quad (5.41)$$

5.5 Identifikovanie lineárneho modelu

Následujúcim cieľom identifikácie bude snaha určiť v Rov. (5.40) jednotlivé koeficienty nachádzajúce sa v maticiach A a B . Jedná sa o to, že v posledných úvahách boli uvedené spôsoby, ako sa na systém možno pozerať, aké fyzikálne procesy sú v rámci systému očakávané a tiež boli uvedené vzťahy a experimenty, na základe ktorých je možné určiť predbežnú hodnotu týchto koeficientov. Avšak treba mať na pamäti, že v procese levitácie zasahujú do systému aj také vstupy, ktoré nie sú súčasťou modelu. Taktiež sme uvažovali niekoľko zjednodušení vrátane linearizácie. Ak by sme aj poznali presnú hodnotu veličín z Rov. (5.40), na základe týchto úprav je veľmi pravdepodobné, že z toho plynúce dátá by nie úplne odpovedali realite. Preto je nutné presnosť týchto hodnôt voči skutočnému systému zlepšiť. Za účelom tohto zlepšenia ďalej vykonáme odhad parametrov lineárneho modelu.

Pri odhade parametrov algoritmus hľadá takú kombináciu hodnôt, aby čo najlepšie vzniknutý model odpovedal určitej referenčnej vzorke – experimentu. Nakoľko sa jedná o nestabilný systém, tento experiment musel byť realizovaný pomocou riadenia v uzavretej slučke. Aby sa dosiahli čo najjustálenejšie hodnoty s čo najmenším šumom, využil som za týmto účelom LQ riadenie s použitím už spomenutého modelu v úvode kapitoly⁸. Dané riadenie nevyužíva na odhad presnej hodnoty stavov žiadnen pozorovateľ alebo filter (je použitá len numerická derivácia pre určenie rýchlosťi), takže použité dátá predstavujú priamo namerané stavy zo senzorov, bez akejkoľvek úpravy.

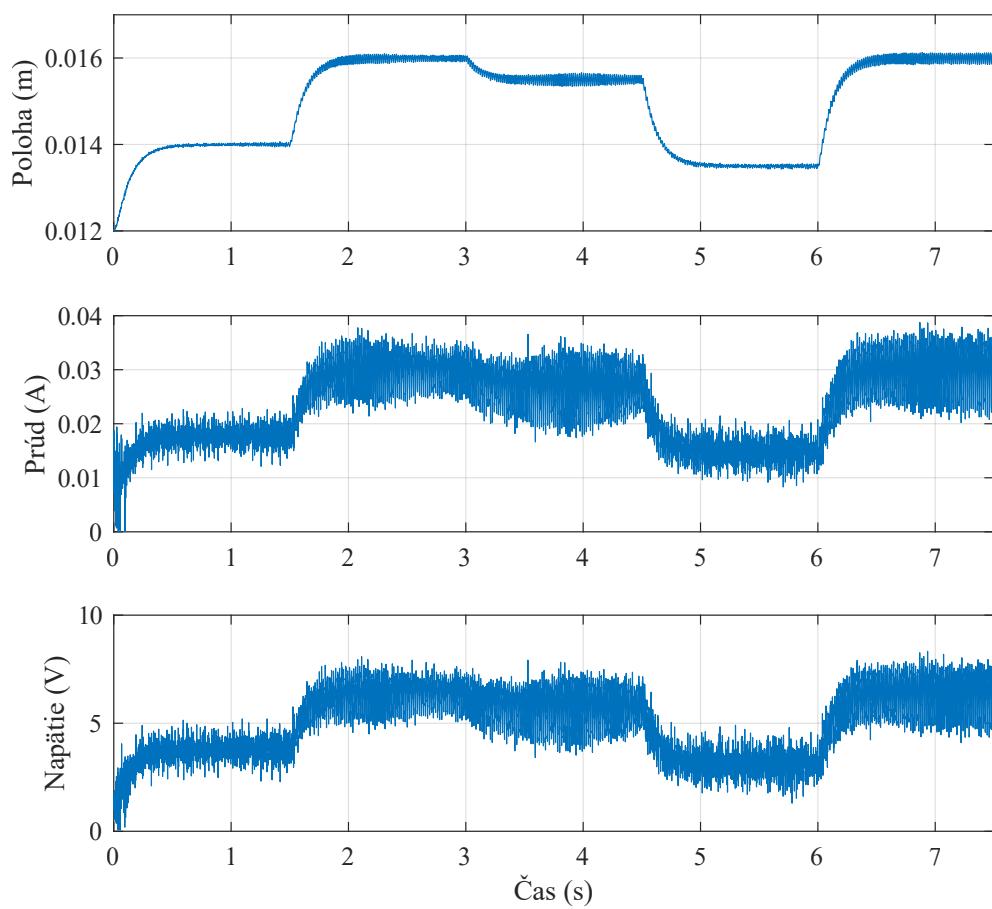
Ako som uviedol, pri zadávaní úlohy odhadu parametrov porovnávame namerané dátá s modelom, ktorý by mal dané chovanie opisovať. Nakoľko je model v otvorenej slučke nestabilný, dochádza ku komplikáciám pri hľadaní parametrov u algoritmov identifikujúcich v čase. Na základe pokusov, ktoré boli pri snahe systém MagnetoShield identifikovať sa dospele k záveru, že ideálnym riešením je identifikácia na základe frekvencií charakterizujúcich namerané dátá. Takúto reprezentáciu signálov dosiahneme procesom transformácie do frekvenčnej oblasti pomocou tzv. Fourierovej transformácie (bližšiemu opisu Fourierovej transformácie, ako aj dôvodom prečo je vhodnejšia na identifikáciu nestabilných systémov sa v tejto práci venovať nebudem, nakoľko sa nejedná o triviálnu problematiku).

Pred použitím diskrétneho algoritmu transformujúceho namerané dátá nazývaného Rýchla Fourierova transformácia (ang. *Fast Fourier Transformation – FFT*), je potrebné dátá jemne upraviť. Ako som už písal, lineárny model prezentovaný v Rov. (5.40) má od nelineárneho modelu odlišné stavy. Rozdiel spočíva v tom, že stavy v lineárnom modeli sú relatívne vzhľadom na linearizačné body. Takže nehovoria, aká veľká je hodnota danej veličiny, ale to, o koľko sa daná veličina odlišuje od hodnoty jej linearizačného bodu. Tým, že merané dátá na Obr. 5.6 sú absolútne merania, správnu formu dostaneme tak, že od nameraných hodnôt jednoducho linearizačné body odčítame ako je zobrazené na Obr. 5.7. Následne je možné aplikovať FFT a dostávame reprezentáciu vo frekvenčnej oblasti charakterizovanú Obr. 5.8.

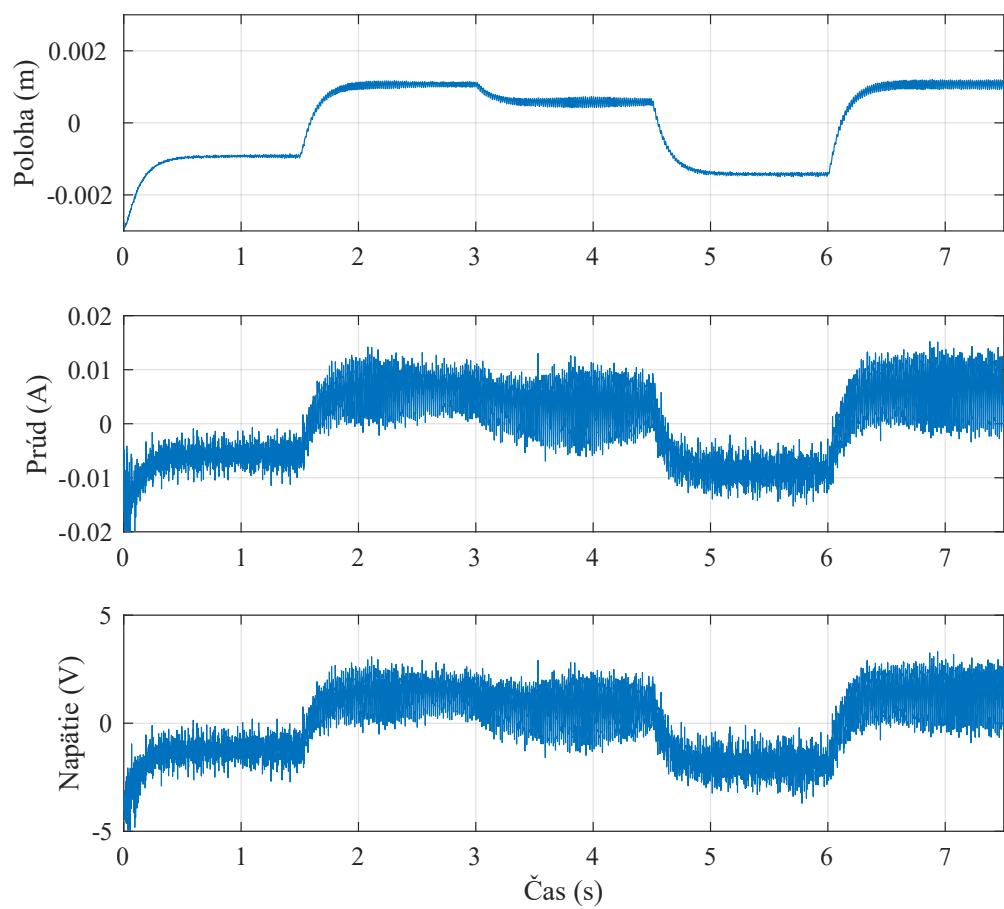
Reprezentácia signálu vo frekvenčnej oblasti zobrazená na Obr. 5.8 má na x-ovej osi vynesenú frekvenciu periodického signálu v jednotke Hertz. Na y-ovej osi je zobrazená magnitúda danej zložky signálu pri konkrétej frekvencii. Táto magnitúda je uvádzaná v jednotke dB (decibel), pri čom prepočet medzi jej jednotkovou veľkosťou y v metroch a veľkosťou v dB je $20\log_{10}(y)$. Toto zobrazenie je na prvý pohľad pomerne nekomfortné a mätúce. Ak sa ale zamyslíme nad spôsobom, aké hodnoty nadobúda y-ová os zistíme, že frekvencie väčšie ako 20 Hz majú príliš malú veľkosť na to aby boli čitateľné bez uvedeného prevodu. Taktiež sa frekvencie končia pri približnej hodnote 150 Hz a pre vyššie frekvencie už zobrazenie transformácie neexistuje (ohraňenie čiernej čiarou vľavo). To je dané vzorkovacou frekvenciou, akou sme realizovali zber dát ($T_s = 0.00325$ s). Na to aby sa pôvodný signál dal z navzorkovaných bodov vierohodne zostaviť, je nutné aby vzorkovacia frekvencia bola aspoň dva krát taká veľká, ako je najväčšia z rekonštruovaných frekvencií (tzv. *Nyquistova frekvencia*) [7, Lecture 21: Sampling].

Táto reprezentácia nám taktiež podáva prvú, ľahko čitateľnú informáciu o chovaní sa systému a jeho spracovávaní signálu. Jedna zo základných vlastností lineárnych časovo-nepremenlivých systémov (*LTI*, teda naša reprezentácia uvedená v Rov. (5.40)) je, že nevedia vytvárať nové frekvencie. Signál ktorý pošleme do systému ako vstup dostaneme aj na výstupe, iba že so zmenenou amplitúdou a istým fázovým posunom [7, Lecture 9: Frequency Response]. Tento fakt

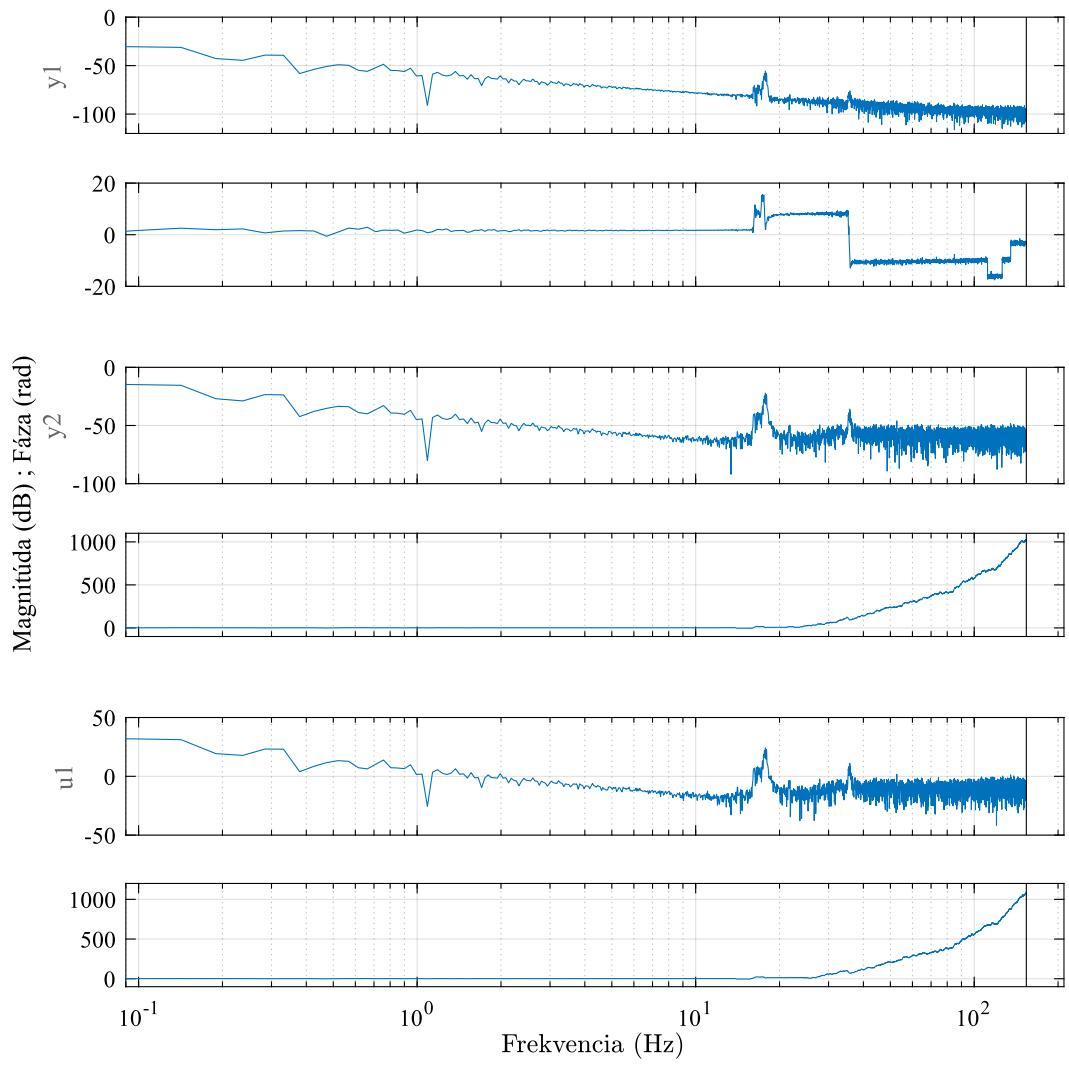
⁸Pri levitácii okrem vertikálneho pohybu, ktorý je modelom opísaný a je našou snahou ho riadiť, dochádza aj k horizontálnym osciláciám. Tie sa prejavujú pri meraní polohy ako istý druh šumu, nakoľko pri každom výkyve do strany sa magnetické pole na Hallovom senzore zmení. Kvalitné riadenie založené na systémovom modeli, akým je LQ je do istej miery schopné tento šum eliminovať. Tým dostávame “čistejšie” dátá pre účel identifikácie.



Obr. 5.6: Priebeh meraných stavov a vstupu.



Obr. 5.7: Priebeh stavov a vstupu po odrátaní linearizačných bodov.



Obr. 5.8: Vstupná veličina a stavové veličiny reprezentované vo frekvenčnej oblasti.

nám pomáha pri priamom porovnávaní jednotlivých frekvencií vstupov a výstupov.

Z Obr. 5.8 vidno, že vstupný signál $u1$ (napätie) má pri nízkych frekvenciach pomerne veľkú amplitúdu. Tá sa mapuje na výstupoch do signálov $y1$ (poloha) a $y2$ (prúd) so značne menšou veľkosťou. To je aj prirodzené nakoľko prúd a poloha sú rádovo na úrovni mili-jednotiek. Zaujímavou však môže byť podobnosť medzi napäťim a prúdom. Nakoľko som vzťah medzi týmito veličinami uvádzal, nepôsobí možno až tak prekvapivo. Ale dá sa považovať za peknú ukážku úzkej závislosti medzi nimi. Nie sú ale úplne proporčne rovnaké. Člen A_{32} (v Rov. (5.40) matica A , 3. riadok, 2. stĺpec) vnáša do mapovania prúdu aj účinok pohybu magnetu, avšak tento efekt je pomerne malý. Preto si možno všimnúť rozdiel len pri špecifických frekvenciách (napr. frekvencie v okolo 20-30 Hz majú na prúd menší vplyv ako pri napäti).

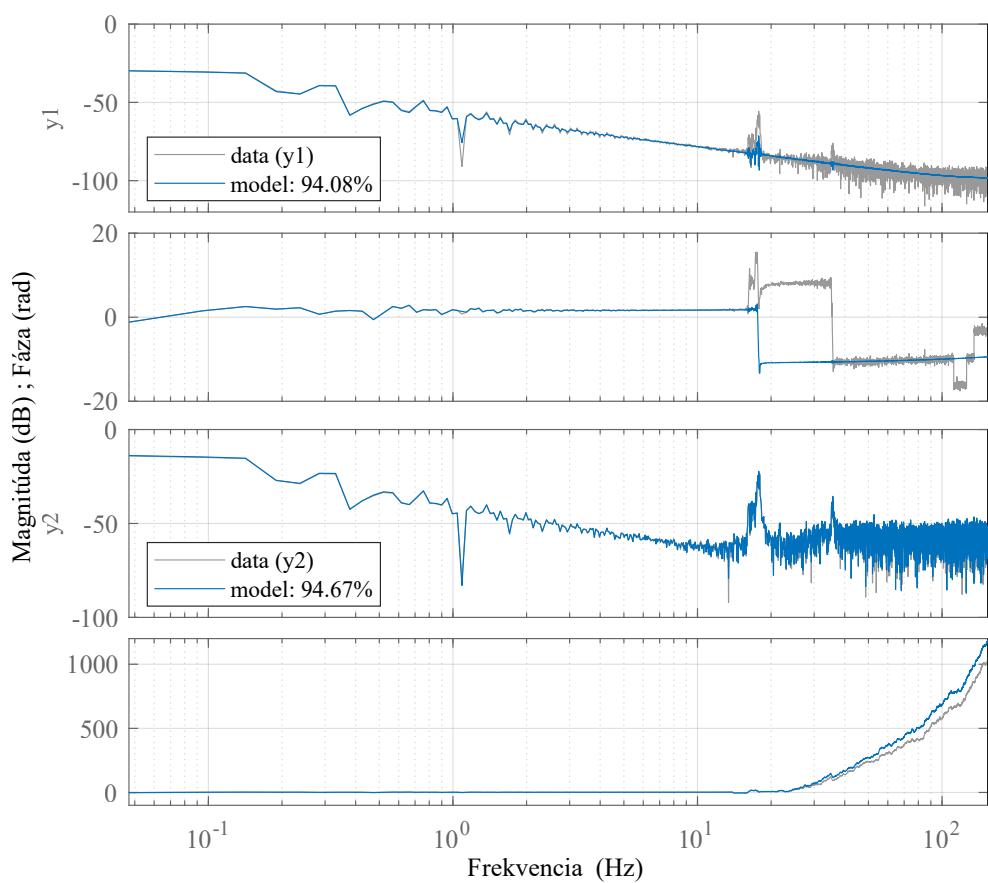
Naopak, pri pohľade na frekvenčnú analýzu signálu charakterizujúceho polohu vidno, že pre malé frekvencie sa mení len magnitúda signálu, tvar ostáva totožný so vstupom. S rastúcou frekvenciou ale ich podobnosť klesá. Neprestajne nižšie hodnoty magnitúdy pre vyššie frekvencie signalizujú, že tieto frekvencie strácajú na význame pri tvorbe výsledného signálu polohy (pre prúd a napätie naopak, magnitúda vysokých frekvencií ostáva približne rovnaká). Ešte badateľnejší rozdiel vidno pri pohľade na fázové posuvy. Zatiaľ čo prúd a napätie majú signály približne rovnako posunuté, pri polohe je vidieť razantne rozdielny vývoj vo fázovom posuve.

Ďalším krokom bude definovanie určitých obmedzení, v rámci ktorých bude identifikovaný model dávať fyzikálny význam. Koeficienty matice A v Rov. (5.39) sú závislé od troch fyzikálnych veličín R , L , m (elektrický odpór, indukčnosť cievky, hmotnosť magnetu), dvoch konštant systému K a c (magnetická konštanta systému, tlmenie) a od linearizačných bodov x_{01} , x_{02} a x_{03} (hodnoty pre polohu, rýchlosť a prúd). Nakoľko som odhad parametrov robil v prostredí MATLAB pomocou funkcie `sses()`, ktorá mení hodnoty koeficientov stavového modelu a nie hodnoty samostatných parametrov⁹, musel som obmedzenia zúžiť len na ohraničenia, či daný koeficient matice A a matice B môže pre túto stavovú reprezentáciu nadobúdať záporné resp. kladné hodnoty. Síce týmto spôsobom nie sme schopní algoritmu úplne zabrániť, aby niektoré parametre nenadobúdali hodnoty mimo ich fyzikálny význam, ale nakoľko linearizačné body sú nemenné hodnoty a hmotnosť poznáme natoľko presne, že jej odhad možno označiť za zbytočný, vieme späťne jednotlivé parametre dopočítať a z toho určiť, či sa ich hodnoty pohybujú v zmysluplnom intervale (táto úvaha je podmienená tým, že sa jedná o *LTI* systém).

Pred spustením procesu identifikácie a odhadu hodnôt koeficientov v maticiach, som vykonal ešte jednu operáciu. Na Obr. 5.9 je porovnanie prvotného modelu (5.41) s nameranými dátami. Číselné hodnoty tohto modelu sú stanovené priamo z procesu modelovania, teda na základe vzorcov a experimentov, ktoré som opísal vyššie. Toto porovnanie nám poskytuje informáciu o tom, nakoľko presné boli naše úvahy o modeli a procesoch, ktoré v ňom prebiehajú. Z porovnania je zrejmé, že model už pred procesom odhadovania koeficientov je veľmi presný a s najvyššou pravdepodobnosťou by sa dal použiť za účelom riadenia. Tento výsledok je obrovským plusom, ktoré iba podporuje možnosť, že mnou vytvorený model je skutočne adekvátna abstrakcia systému.

Po spustení identifikačného skriptu ([19, možno nájsť v časti *examples/MagnetoShield/AlternativeModel*]) dostaneme odhad jednotlivých koeficientov stavového systému a aj jeho grafické porovnanie s nameranými dátami na Obr. 5.10. Porovnanie hodnôt konkrétnych parametrov modelu, ktoré som v rámci modelovania odhadol a následne pomocou identifikačných funkcií prepočítal, možno vidieť v tabuľke 5.3. Taktiež sú v tabuľke uvedené hodnoty parametrov, ktoré neboli cieľom odhadu, ale uvádzam ich pre celistvosť informácie. Celý výsledok identifikácie je

⁹Koeficientom matice sa myslí prvok matice umiestnený na pozícii a_{ij} , kde a je matica, i je riadok a j je stĺpec matice.



Obr. 5.9: Porovnanie modelu systému s nameranými dátami pred odhadom parametrov.

Veličina	Prvotný odhad	Identifikovaná hodnota
L – Indukčnosť cievky	0.2350 H	0.3907 H
R – Odpór	210.0 Ω	213.5237 Ω
K_m – Konštantá K pre mechanickú rovnicu	$1.5307 * 10^{-9}$ N·A $^{-1}$	$5.9752 * 10^{-9}$ N·A $^{-1}$
K_e – Konštantá K pre elektrickú rovnicu	$1.5307 * 10^{-9}$ N·A $^{-1}$	$4.4449 * 10^{-7}$ N·A $^{-1}$
c – Tlmenie	0 kg·s $^{-1}$	0 kg·s $^{-1}$
m – Hmotnosť magnetu	$0.74 * 10^{-3}$ kg	
x_{01} – Linearizačný bod pre polohu	0.0149 m	
x_{02} – Linearizačný bod pre rýchlosť	0 m·s $^{-1}$	
x_{03} – Linearizačný bod pre prúd	0.0236 A	

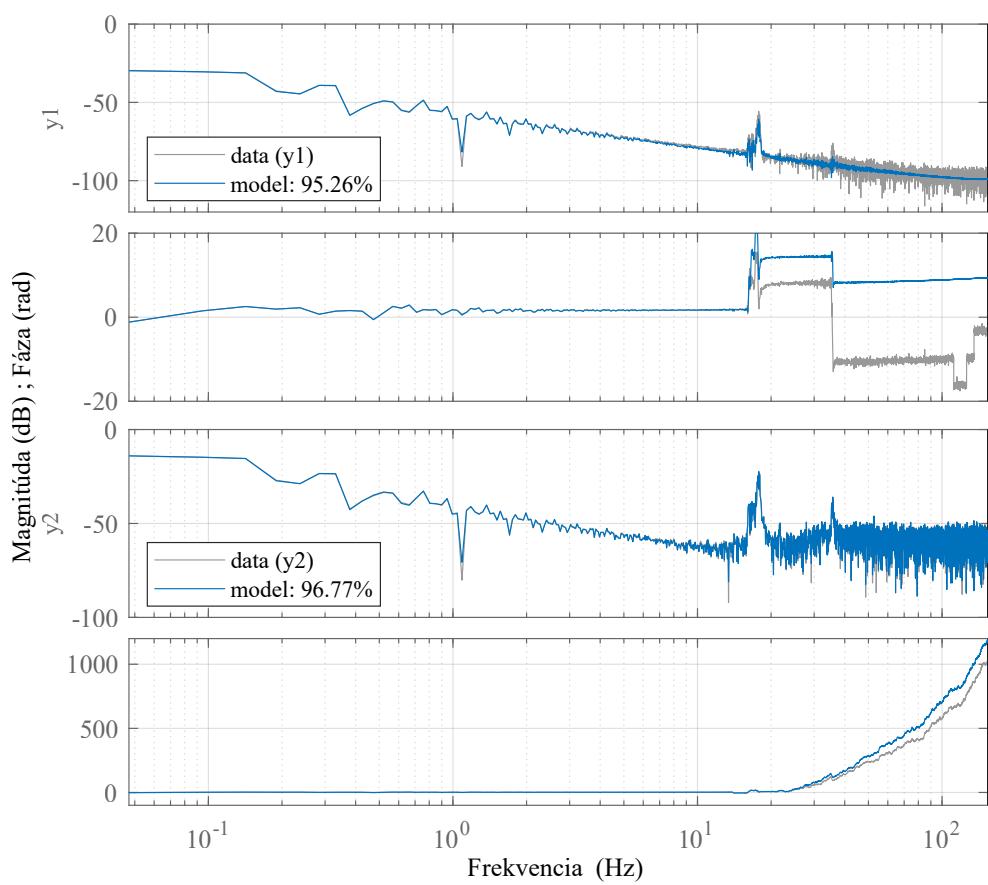
Tabuľka 5.3: Hodnoty parametrov lineárneho modelu.

uložený v modeli, ktorý je výstupným argumentom funkcie `ssest`. Tu je dôležité poznamenať, že identifikovaný model neobsahuje len nové matice identifikovaného stavového priestoru, ale aj rôzne informácie o nepresnostiach a kvalite modelu, ktoré môžu byť následne využité pri spracovaní dát a ladení regulátorov resp. iných algoritmov spojených so systémom.

Takže Obr. 5.10 reprezentuje porovnanie nameraných dát a modelu. Jedná sa o frekvenčné zobrazenie výstupných signálov nameraných na senzoroch (šedá – ten istý priebeh ako je na Obr. 5.8) a o signály vytvárané modelom (modrá). Obe sú odozvou systému na totožný vstupný signál. Elektrická odozva (závislosť prúdu od napäcia) sa chová rovnako na celom intervale frekvencií. Pri časti opisujúcej polohu pri vysokých frekvenciách má odozva z modelu menšiu variáciu vo veľkosti amplitúdy. Nakoľko ale stredná hodnota kopíruje merané dátu, možno tieto vysokofrekvenčné komponenty označiť za šum vzniknutý pri meraní a neopísateľné faktory v modeli. Amplitúda týchto frekvencií je taktiež malá a teda aj ich vplyv na výsledok nie je až taký zásadný. Čo sa týka fázového posuvu, model vystihuje charakter zmeny fázy aj keď dochádza k značným rozdielom medzi nameranými a modelovými dátami. Fáza ale v prípade takéhoto signálu nehrá až takú zásadnú úlohu. Opäť tu platí, že nízke frekvencie pri modeli sa zhodujú s nameranými dátami. S klesajúcou amplitúdou nevyhnutnosť zhody oboch fáz ale klesá. Nakoľko ak sú amplitúdy veľmi malé a frekvencie vysoké, tak aj nesprávny posuv sa pri sčítaní signálov prejaví minimálne.

V tab. 5.3 sú uvedené dve varianty konštanty K a to K_m a K_e . V rámci kontextu tejto kapitoly by táto úvaha mala byť nesprávna, nakoľko by sa malo jednať o tú istú hodnotu, tej istej konštanty. Je ale potrebné vziať do úvahy fakt, že táto konštantá má reprezentovať magnetické vlastnosti oboch magnetov, ktoré sú závislé od ich materiálu, ako aj tvaru. A práve tieto informácie sú pomerne ľahko merateľné a zohľadniteľné v zmysle zjednodušení, ktoré sme zaviedli. Taktiež hodnota konštanty je veľmi malá a nakoľko sa nachádza v čitateli, akceptovanie určitej možnosti variácie nám dáva omnoho presnejší model pri približnom zachovaní významu samotného parametra. V prípade snahy striktne dodržať opísanú logiku, je možné využiť hodnoty parametrov pred identifikáciou, ktoré sa preukázali taktiež ako presné (Obr. 5.9).

V pôvodnom návrhu modelu v Rov. (5.41) si možno všimnúť, že konštantu tlmenia c uvažujeme rovnú 0. Bolo to preto, lebo sme nemali žiadnu relevantnú informáciu o tomto parametri a jeho existencia v modeli bola len predpokladaná. Zaujímavým výsledkom odhadovania parametrov ale je, že po identifikácii sa hodnota tlmenia c nezmenila (tab. 5.3). Dokonca v priebehu experimentov výsledky naznačovali možnosť $c < 0$, čo by znamenalo že dochádza k istému neopísanému zrýchľovaniu závislého na momentálnej rýchlosťi. Je pomerne pravdepodobné, že to nejako súvisí so skutočnosťou, že za účelom levitácie využívame permanentný a silný magnet. Nakoľko to ale môže byť spôsobené aj inými faktormi, záver že konštantá $c = 0$ je opodstatnenejší.



Obr. 5.10: Porovnanie identifikovaného modelu s nameranými dátami.

Ak vezmeme do úvahy uvedenú argumentáciu spolu s vysokou percentuálnou zhodou modelu po (Obr. 5.10), ale aj pred identifikáciou (Obr. 5.9), môžeme skonštatovať, že uvedený opis modelu je veľmi podobný skutočnému chovaniu sa systému a možno ho využívať na návrhy a simuláciu riadenia. Výsledná reprezentácia systému po zaradení identifikovaných koeficientov do modelu vyzerá takto:

$$\begin{aligned} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \\ \Delta x_3(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 993.6622 & 0 & -162.7508 \\ 0 & 22.9300 & -546.5014 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \\ \Delta x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 2.5594 \end{bmatrix} \Delta u(t), \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \\ \Delta x_3(t) \end{bmatrix}. \end{aligned} \quad (5.42)$$

V rámci Kap. 5 som teda uviedol dva modely reprezentujúce systém MagnetoShield. V prvom prípade sa jednalo o model uvedený v článku venovanom MagnetoShieldu R2 založený na práci Wen Yuho a Xiaou Lina „A Magnetic Levitation System for Advanced Control Education“ [28]. Druhý model navrhnutý mnou je abstrakciou systému, založenou na principiálne inom náhľade na interakciu magnet-elektrický magnet. Obidva modeli sa ale preukázali použiteľné a systém za účelom riadenia veľmi dobre opisujúce. V nasledujúcich kapitolách budem uvádzať jednotlivé spôsoby riadenia využívajúce mnou navrhnutý model. Ale nakoľko sa jedná o riadenia používajúce *LTI* reprezentáciu systému, uvedené postupy a spôsob návrhu riadenia je možné aplikovať na akýkoľvek lineárny časovo nemenný model MagnetoShieldu.

6 Riadenie

V rámci tejto kapitoly na úvod uvediem teoretické poznatky približujúce všeobecné fungovanie riadiacich algoritmov vo spätej väzbe založených na stavovom modeli. Okrem tejto problematiky priblížim aj spôsob, ako sa vysporiadať s tým, ak v systéme nedokážeme merať priamo všetky potrebné veličiny. Následne sa posnažím demonštrovať využitie opísanej teórie pri návrhu a aplikácii riadiacich algoritmov na reálny systém MagnetoShield pri sledovaní vopred definovanej trajektórie.

V rámci projektu bolo vytvorených viacerо rozhraní (Kap. 4) a sice princípy algoritmov sú v každom rozhraní rovnaké, jednotlivé prostredia sa od seba líšia tým, ako zdrojový kód spracovávajú a od toho sa aj odvíja čas behu riadiaceho programu. To sa prejavuje maximálnou vzorkovacou frekvenciou pre konkrétnе rozhranie a pre danú prototypizačnú dosku. Taktiež sa to môže prejaviť do istej miery aj v ladení regulátora. Počas mojej diplomovej práce som používal predovšetkým Arduino DUE, ale softvérové rozhrania Arduino IDE a Simulink boli navrhované aj pre dosky UNO a MEGA. Čo sa týka samotných rozhraní, moja práca sa zameriava predovšetkým na využitie Simulinku, ktorý bol hlavný vývojový program počas tejto diplomovej práce. Z týchto dôvodov, v tabuľke 6.1 uvádzam odporúčané vzorkovacie periódy pre jednotlivé prototypizačné dosky Arduino v rámci rozhrania Simulink. Tieto vzorkovacie časy slúžia ako približná referencia uvádzajúca, ako rýchlo vedia jednotlivé programy na daných doskách fungovať a taktiež sú to časy použité pri mnou zostavených experimentoch opísaných nižšie. Každopádne, je ale možné chod každého programu zrýchliť v závislosti od použitých funkcií a miery komplexnosti programu.

Odporúčané vzorkovacie časy			
Riadiaci algoritmus	Arduino UNO	Arduino MEGA	Arduino DUE
PID	2.0 ms	2.0 ms	1.0 ms
Uloženie pôlov	2.5 ms	2.5 ms	1.5 ms
LQR	2.5 ms	2.5 ms	1.5 ms
MPC	-	-	3.0 ms
eMPC	-	-	3.0 ms

Tabuľka 6.1: Vzorkovacie časy vzhľadom na riadenie a prototypizačnú dosku.

Taktiež, ako som v úvode Kap. 5 naznačil, v rámci projektu sú využívané dva modeli opisujúce systém. Príklady sú na stránke projektu demonštrované použitím prvotne navrhnutého modelu z článku [20]. Najbližšie podkapitoly vysvetľujú princípy za týmito príkladmi, avšak v tejto práci uvádzam výsledky z použitia mnou navrhnutého modelu z Kap. 5, s čím sa môžu spájať určité rozdielnosti v ladení regulátorov, vzorkovacích časoch a sledovaných referenciach¹. Princíp a uvedené pravidlá sú ale totožné pre oba modeli (ladenie je tiež podobné). Takisto je nutné poznamenať, že permanentný magnet využívaný pri riadení a identifikácii v príkladoch dostupných na stránke projektu [19] je iný ako ten, ktorý som používal počas mojej diplomovej

¹Na stránke možno nájsť aj príklady pre použitie tohto modelu v časti *MagnetoShield_AlternativeModel_Examples*.

práce. V tabuľke 3.1 sú uvedené oba magnety a možno ich chovanie sa v systéme považovať za, do istej miery, rovnaké. Napriek tomu, osobne odporúčam využívanie 8 mm magnetu v spojení s modelom opísaným v Kap. 5, nakoľko bol tejto kombinácií venovaný dôkladnejší výskum.

6.1 Stavové spätnoväzobné riadenie

Kap. 5 objasňovala spôsob, ako reálny systém opísť pomocou matematicko-fyzikálnych vzťahov. Výsledkom bolo vytvorenie reprezentácie v tzv. stavovom priestore. Otázkou ale ostáva, k čomu sú táto reprezentácia a model celkovo dobré a ako vieme túto znalosť modelu využiť za účelom riadenia.

Najzákladnejšou otázkou, ktorou sa môžeme v prípade riadenia systému MagnetoShield zaoberať je, či je systém stabilný. Budť na základe predošlého textu alebo aj vlastným úsudkom si vieme jednoducho dokázať, že tento systém prirodzene stabilný nie je. Ak umiestníme levitujúci magnet do nejakej výšky (okrem uloženia na zem), magnet okamžite začne padať. V prípade, že zapneme elektromagnet, magnet sice nepadne, ale pre zmenu sa hnedť prilne k vrchnému ohraďeniu priestoru v ktorom má lietať. Takže konštantným alebo nulovým vstupom do systému nevieme magnet prinútiť ostať v inej ako hraničnej pozícii. Táto vlastnosť systému je ľahko opísateľná pomocou tzv. pólov.

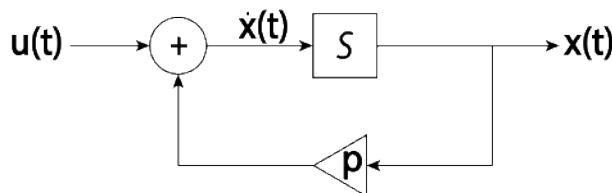
Pre lepšie pochopenie, čo to pól je a ako ovplyvňuje systém, uvažujme jednoduchú diferenciálnu rovnicu

$$\dot{x}(t) = px(t) + u(t). \quad (6.1)$$

Alternatívnym zobrazením tohto vzťahu by bolo prekresliť ho do schématického zobrazenia, ako je na Obr. 6.1. Pri takomto vyjadrení systému je prirodzenejšie o hodnotách a riešení uvažovať ako o signále. Okrem vstupu $u(t)$ a výstupu $x(t)$ tu vidíme blok S nazývaný integračný operátor (integruje, alebo tiež akumuluje signál – nie je tým myšlený Laplaceov obraz integrácie $1/s$, ale operátor obsahujúci túto operáciu), sumu spájajúcu vstupný signál so spätnou väzbou (výstupom) a násobiteľ p , ktorým výstupný signál v spätej väzbe násobíme.

Teraz, iba v tendenciach Obr. 6.1, uvažujme, ako sa bude vyvíjať výstup $x(t)$ v prípade, že $u(t) = \delta(t)$, teda na vstupe je Diracov impulz. Nakoľko signál prechádza cez integrátor, v čase $t = 0$ dostávame výstup $x(t) = 1$. Vďaka spätej väzbe sa táto hodnota vracia cez násobiteľ p a opäťovne je zintegrovaná. Na výstupe zrazu dostávame funkciu, ktorej hodnota v čase exponenciálne rastie s faktorom p . Je zrejmé, že hodnota p bude rozhodovať o tom, či výstup $x(t)$ pôjde s časom do 0 alebo ∞ a tento násobiteľ sa nazýva pól systému.

Pre detailný obraz o tom, čo to znamená, že násobiteľ p rozhoduje o chovaní sa systému vyzadíme si Obr. 6.1 ako funkcionál systému G , kde funkcionál X bude odpovedať $x(t)$ a funkcionál



Obr. 6.1: Grafické znázornenie diferenciálnej rovnice.

U odpovedá $u(t)$:

$$\begin{aligned} X &= S(pX + U), \\ X(1 - pS) &= SU, \\ G &= \frac{X}{U} = \frac{S}{1 - pS}. \end{aligned} \tag{6.2}$$

Dostávame polynomiálne vyjadrenie prenosu funkcie, charakterizujúce ako sa vstup mapuje na výstup. Aby sme dostali informáciu o tom, ako vyzerá priebeh závislý od času, polynom podelíme:

$$\frac{X}{U} = \frac{S}{1 - pS} = S(1 + pS + p^2S^2 + p^3S^3 + \dots) \tag{6.3}$$

a získaný vzťah môžeme opäťovne použiť pre vyjadrenie signálov v čase tým, že funkcionálne premenné X a U nahradíme funkiami času $x(t)$ a $u(t) = \delta(t)$ a operátor S nahradíme integráciou:

$$\begin{aligned} \frac{x(t)}{u(t)} &= S(1 + pS + p^2S^2 + p^3S^3 + \dots), \\ x(t) &= S(1 + pt + \frac{1}{2}p^2t^2 + \frac{1}{6}p^3t^3 + \dots)\delta(t), \\ x(t) &= e^{pt}v(t). \end{aligned} \tag{6.4}$$

Nekonečná postupnosť polynómov reprezentuje exponenciálnu funkciu, kde $v(t) = S\delta(t)$, teda integrál Diracovho impulzu $\delta(t)$, predstavuje jednotkový skok². Takže vidíme, že to či bude výstupný signál $x(t)$ konvergovať alebo divergovať závisí od násobiteľa – pólu p [7, Lecture 4: Continuous-Time (CT) Systems], takže

$$\lim_{t \rightarrow \infty} x(t) = \begin{cases} 0 & \text{ak } p < 0, \\ \infty & \text{ak } p > 0, \\ v(t) & \text{ak } p = 0. \end{cases} \tag{6.5}$$

Dôležitým poznatkom ale je, že sme uvažovali o spojitém CT (*continuous-time*) systéme. Veď mi podobný postup možno aplikovať aj pri úvahách o diskrétnom systéme – DT (*discrete-time*). Uvažujme diferenčnú rovnicu ktorá by bola alternatívou k (6.1) v DT,

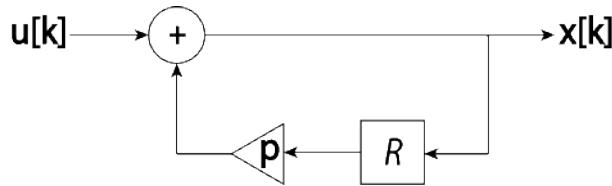
$$x[k] = u[k] + px[k - 1]. \tag{6.6}$$

Na Obr. 6.2 je zobrazená graficky. Blok R tzv. operátor posunu má v tomto prípade úlohu signál oneskoríť o jednu vzorku (signál $x[k]$ sa dostane na sumu až v ďalšej vzorke $k + 1$). Opäť, ak budeme uvažovať nad tým, ako ovplyvňuje p vývoj výstupu po jednotkovom impulze na vstupe, zistíme, že pre isté hodnoty bude konvergovať k 0 a pre iné pôjde do ∞ a pre $p = 1$ ostane konštantný.

To sa dá interpretovať aj číselne, opäť pomocou vyjadrenia systémového funkcionálu:

$$\begin{aligned} X &= pRX + U, \\ X(1 - pR) &= U, \\ G &= \frac{X}{U} = \frac{1}{1 - pR}, \end{aligned} \tag{6.7}$$

²V literatúre sa jednotkový skok alebo skoková zmena vstuпу často uvádzá ako $u(t)$.



Obr. 6.2: Grafické znázornenie diferenčnej rovnice.

kde po delení dostávame

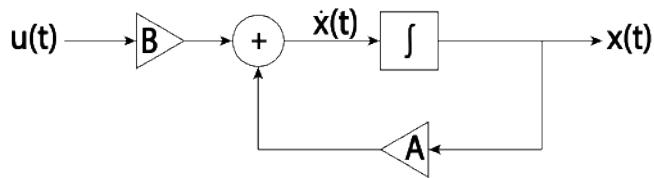
$$\frac{X}{U} = \frac{1}{1 - pR} = 1 + pR + p^2R^2 + p^3R^3 + \dots \quad (6.8)$$

Operátor R a jeho mocniny značia oneskorenie signálu o veľkosť exponenta ($R^2 \leftrightarrow [k-2]$), takže prenos signálu sa dá vyjadriť ako postupnosť p^k (polynomiálne vyjadrenie Rov. (6.8) je vyjadrením tejto exponenciálnej postupnosti).

Z toho je priamo vidieť, akým spôsobom pól p ovplyvňuje stabilitu DT systému [7, Lecture 3: Feedback, Poles, and Fundamental Modes].

$$\lim_{k \rightarrow \infty} x[k] = \begin{cases} 0 & \text{ak } p < |1| \\ \infty & \text{ak } p > |1| \\ 1 & \text{ak } p = 1 \\ \pm 1 (\text{osculuje}) & \text{ak } p = -1 \end{cases} \quad (6.9)$$

Toto je priama ukážka toho, čo je pól a akým spôsobom od neho závisí stabilita systému. Ak sa teraz pozrieme na Rov. (5.39) v rovnici dynamiky vidíme veľmi podobnú štruktúru, ako je tá, v ukážkovom príklade vyššie pre CT Rov. (6.1) (to isté platí pre DT formu stavového priestoru a uvedený príklad). Jediným rozdielom je to, že miesto násobiteľa p , ktorý reprezentoval jediné číslo, máme zrazu maticu A a vstupom nie je iba signál $u(t)$, ale je násobený určitým faktorom v podobe matice B (viď Obr. 6.3). Okrem tohto, princíp ostáva rovnaký, opäť o stabilite budú rozhodovať póly systému nachádzajúce sa v matici A tiež označované ako vlastné čísla matice.



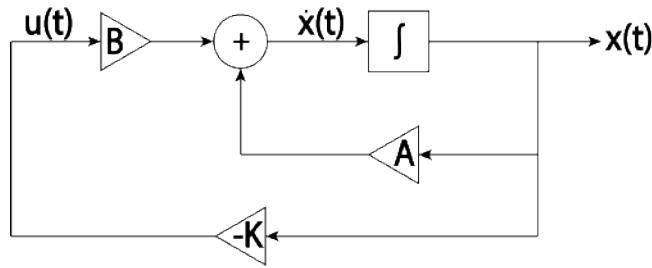
Obr. 6.3: Grafické zobrazenie rovnice dynamiky stavovej reprezentácie.

Kedže je systém MagnetoShield nestabilný, s dynamikou matice A nedokážeme nič urobiť. Časť ale, na ktorú máme dosah je vstupná veličina $u(t)$. Na Obr. 6.4 je znázornená schéma systému v prípade, že zo vstupného signálu $u(t)$ urobíme funkciu výstupu. Tento vzťah definujúci vstupný signál sa nazýva riadiaci zákon:

$$u(t) = -Kx(t). \quad (6.10)$$

Záporná spätná väzba súvisí s našou snahou systém stabilizovať – zmenšiť jeho stavy a dosiahnuť $x(\infty) = 0$. Po takejto substitúcii možno systém diferenciálnych rovníc upraviť spôsobom,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ \dot{x}(t) &= (A - BK)x(t). \end{aligned} \quad (6.11)$$



Obr. 6.4: Schéma zápornej spätej väzby stavového modelu.

A tu vidíme, že zrazu dostávame nový systém charakterizovaný novou maticou $(A - BK)$, ktorého póly už vieme ovplyvniť konštantou K , ktorej voľba je na nás.

Totožnú logiku je možné aplikovať aj pre DT. Avšak pri použití DT je nutné upraviť matice A a B (matica C a ostane hodnotou rovnaká, ako aj D ale tú neuvažujeme) [22, Kap. 2.3]:

$$\begin{aligned} A_d &= e^{AT_s}, \\ B_d &= \int_0^{T_s} e^{A\tau} d\tau B, \\ C_d &= C, \\ D_d &= D. \end{aligned} \tag{6.12}$$

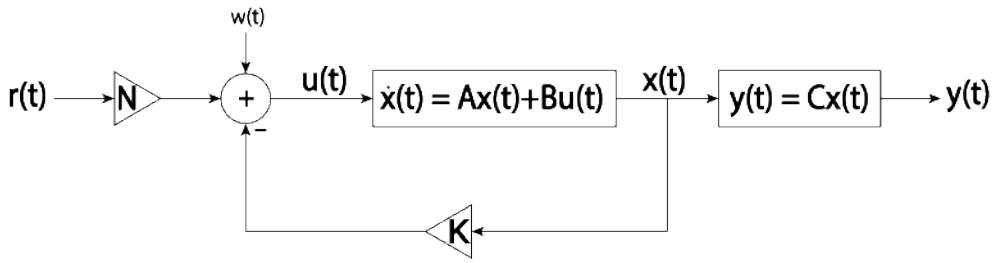
Ako vidíme, hodnoty matíc budú závisieť od nami zvolenej vzorkovacej periódy T_s , preto je nutné pri návrhu systému vziať do úvahy aj tento aspekt. Preto je ladenie regulátora vždy špecifické pre daný systém pri určitom vzorkovaní.

Tieto úvahy sú základom fungovania spätnoväzobných riadení založených na stavovej reprezentácii systému, ako je riadenie uložením pôlov alebo LQ regulátor.

6.2 Sledovanie referencie

So spätnoväzobným riadením a uvedenou teóriou sa spája ešte jeden problém. Záporná spätná väzba sa snaží riadiť stavy do nuly (ak je $x(t) > 0$ prejaví sa to zmenšením vstupu, ak je $x(t) < 0$ vstup sa naopak zväčší³). Obvyklou úlohou riadenia je dosiahnuť žiadanú hodnotu stavov, ktorá sa môže od nuly lísiť. Z Obr. 6.4 vyplýva, že stabilizovaný systém bude v uvedenom zapojení produkovať zo spätej väzby nulový vstup $u(t)$ (stabilizovaným sa myslí, že stavy sú nulové). Ak by sme k tomuto vstupu pripočítali hodnotu $r(t)$, opäť sa systém bude snažiť produkovať nulovú spätnú väzbu, ale tá už bude odpovedať nenulovému vstupu. Ináč povedané, stabilizovaný systém bude mať konštantný vstup $u(t) = r(t)$ na základe čoho možno očakávať aj nenulovú, konštantnú hodnotu stavov $x(t)$. Nakoľko vstupná hodnota býva často reprezentovaná rozdielnou fyzikálnou veličinou ako sú stavové veličiny, je nutné referenciu $r(t)$ násobiť prevodovým vzťahom (referencia často býva udávaná v jednotkách, v ktorých je sledovaná stavová veličina, ktorú sa snažíme riadiť). Nakoľko je MagnetoShield reprezentovaný ako LTI, tento prevodový vzťah bude v podobe časovo nezávislého násobku N . Výsledný riadiaci reťazec možno vidieť na Obr. 6.5.

³Toto je všeobecná teória, pri aplikácii na MagnetoShield to platí obrátene.



Obr. 6.5: Sledovanie referencie pomocou násobiteľa N .

Okrem spomenutého, možno na Obr. 6.5 vidieť ešte jeden vstup $w(t)$ predstavujúci šum. Uvedené riešenie bude fungovať správne, len ak hodnota tohto šumu bude nulová. Avšak v prípade akejkoľvek inej hodnoty sa budú stavové veličiny od referencie $r(t)$ lísiť práve o hodnotu $w(t)$. Nakoľko je toto riešenie jednoduché a ľahko implementovateľné, dá sa využívať pri simulácii systémov, v ktorých sú hodnoty šumu nulové. Pre použitie v reálnom systéme, akým je MagnetoShield, budeme musieť tento princíp ešte rozšíriť.

Okrem šumu $w(t)$, nazývaného tiež procesný šum, sa môžu náhodné nepresnosti objaviť aj pri meraní stavov. Tie sú reprezentované šumom merania $v(t)$. Aby sme vplyv oboch náhodných veličín eliminovali, pridáme ešte jednu spätnú väzbu z meraného stavu zobrazenú na Obr. 6.6. Rozdielom referencie a meraného stavu získavame hodnotu, ako ďaleko sa od referencie nachádzame. Hneď za tým je na Obr. 6.6 blok reprezentujúci integrátor, akumulujúci tento rozdiel. Práve integrátor zabezpečuje to, že vstup do systému sa bude neprestajne meniť pokiaľ nebude splnená podmienka $y(t) = r(t)$. Tako dostávame nový stav systému $x_i(t)$, opisujúci dynamiku rozdielu žiadanej a nameranej hodnoty riadeného stavu a ďalej ho budeme nazývať integračný stav. Tu vidíme, že násobiteľ N zrazu zohráva totožnú úlohu, akú má násobiteľ K a teda upravuje póly tohto novo vzniknutého stavu a stabilizuje ich [30]. S tým sa spája aj zmena znamienka, nakoľko sa jedná o stav, ktorý sa snažíme uregulovať do nuly.

Vytvorenie nového stavu systému v sebe nesie aj nový rozšírený stavový model (angl. *augmented form*) získaný z rovníc

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ \dot{x}_i(t) &= r(t) - Cx(t), \\ y(t) &= Cx(t). \end{aligned} \tag{6.13}$$

Po ich sčítaní získavame systém:

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix} &= \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t), \\ y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}. \end{aligned} \tag{6.14}$$

Okrem rozšírenia systému treba vziať do úvahy aj rozšírenie riadiaceho zákona, ktoré v sebe zahrňa aj konštantu N násobiaci nový stav $x_i(t)$. Nový riadiaci zákon sa dá vyjadriť ako

$$u(t) = -[K \ N] \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}, \tag{6.15}$$

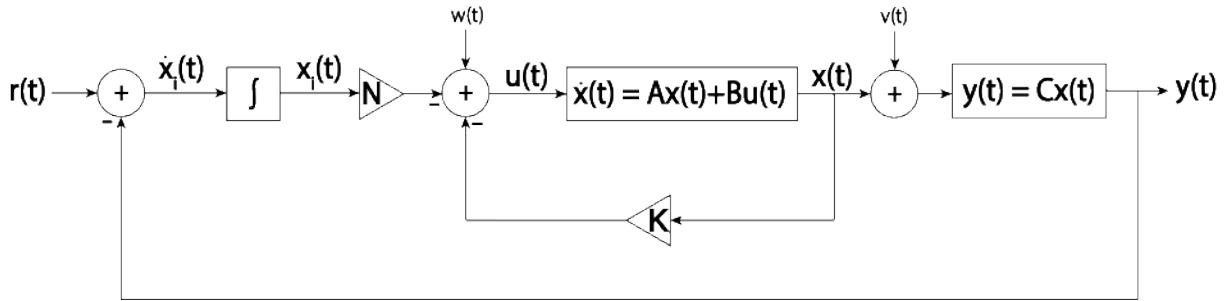
na základe čoho dostávame stavovú reprezentáciu systému z Obr. 6.6:

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix} &= \begin{bmatrix} A - BK & -BN \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(t), \\ y(t) &= [C \ 0] \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}. \end{aligned} \quad (6.16)$$

Nakoľko pri aplikácii budeme využívať DT v skratke uvediem ako sa zmení Rov. (6.14). Ak vykonáme prevod do diskrétnej formy, výsledná rozšírená reprezentácia bude

$$\begin{aligned} \begin{bmatrix} x[k+1] \\ x_i[k+1] \end{bmatrix} &= \begin{bmatrix} A_d & 0 \\ -C_d & 1 \end{bmatrix} \begin{bmatrix} x[k] \\ x_i[k] \end{bmatrix} + \begin{bmatrix} B_d \\ 0 \end{bmatrix} u[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r[k], \\ y[k] &= [C_d \ 0] \begin{bmatrix} x[k] \\ x_i[k] \end{bmatrix}. \end{aligned} \quad (6.17)$$

kde matice A_d, B_d a C_d sú matice stavového priestoru prevedené do diskrétneho tvaru na základe (6.12). Implementácia riadiaceho zákona je totožná ako pri CT v Rov. (6.15), len sa spojité veličiny zmenia na diskrétne $x[k]$ a $x_i[k]$.



Obr. 6.6: Sledovanie referencie pomocou integračného stavu $x_i(t)$.

6.3 Odhad stavov

Pred tým, než sa dostanem k opisu samotných algoritmov by som rád priblížil ešte jednu nemenej podstatnú časť stavovej reprezentácie, o ktorej som zatial veľa nevravel. V Rov. (5.39) je okrem systému opisujúceho chovanie sa MagnetoShieldu aj rovnica definujúca, ktoré stavy sú na MagnetoShielde merateľné, $y(t) = Cx(t)$. Túto časť som doteraz nebral príliš do úvahy, ale nakoľko sa blížime k opisu a implementácii riadenia na reálny systém, je potrebné zamyslieť sa nad tým, aké hodnoty stavov vieme zo systému získať.

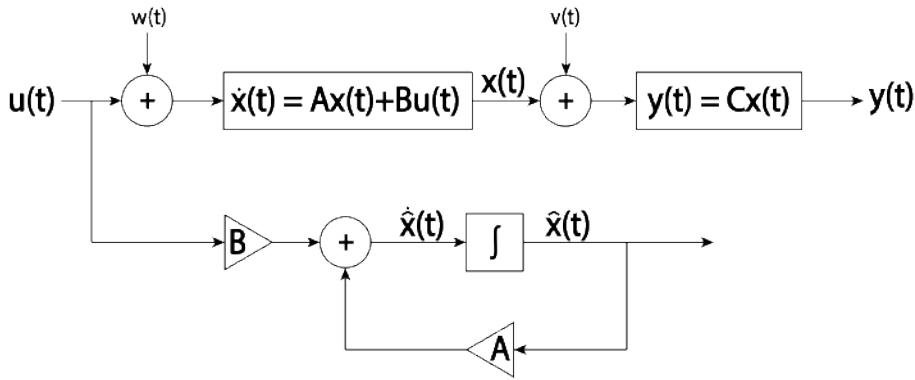
Či už na základe spomenutej rovnice alebo na základe Kap. 3 vieme, že MagnetoShield má zabudované periférie na meranie polohy levitujúceho magnetu a na meranie elektrického prúdu prechádzajúceho elektromagnetom. Tieto dve stavové veličiny vieme teda merať. Ako ale získame informáciu o nemerateľnom stave – rýchlosťi?

Na základe zákonov mechaniky vieme, že ak poznáme polohu v čase, odvodiť si hodnotu rýchlosťi nie je náročná úloha. Veľkosť rýchlosťi vieme v spojitom čase získať ako

$$v(t) = \frac{dh(t)}{dt}, \quad (6.18)$$

kde $v(t)$ je veľkosť rýchlosťi v čase t a $h(t)$ je poloha. Nakoľko naša implementácia riadenia bude prebiehať v DT, toto vyjadrenie môže mať formu

$$v[k] = \frac{h[k] - h[k-1]}{T_s}, \quad (6.19)$$



Obr. 6.7: Koncept pozorovateľa a získania odhadu stavov \hat{x} .

kde n predstavuje k-tú vzorku a T_s je veľkosť vzorkovacej períody. Okrem diskretizačnej metódy v Rov. (6.19) nazývanej *Spätná Eulerova metóda* existuje ešte niekoľko numerických metód aproximujúcich derivácie (dopredná Eulerova, lichobežníková metóda, ...), líšiacich sa v spôsobe mapovania pólov z s-roviny do z-roviny (pre viac informácií viď [7, Lecture 7: Discrete Approximation of Continuous-Time Systems]). Takýto spôsob odhadu ale používať pri práci so Simulinkom nebudeme, pokiaľ to nebude absolútne nevyhnutné, nakoľko je neuveriteľne citlivý na šum.

Pre uvedený príklad je vzťah, ako na základe jedného stavu získať približné hodnoty iného stavu, naozaj jednoduchý. Čo by sa ale dialo v prípade, že by meranie polohy bolo veľmi nepresné prípadne silno zašumené? V takom prípade by výsledná hodnota rýchlosť mala značne inú hodnotu oproti realite. Rovnako sa môžeme zamyslieť, ako by sme získali informáciu o prúde v prípade, že by sme k meraniam prúdu na MagnetoShielde nemali prístup?

Nakoľko poznáme model systému a možno tento model považovať za pomerne presný, najjednoduchším riešením je odhad hodnoty stavu pomocou neho. Intuitívne riešenie by bolo, rovnaký vstup aký posielame do reálneho systému, posieláť do virtuálne simulovaného modelu – pozorovateľa, a výstup $\hat{x}(t)$ použiť miesto merania. Tento koncept je zobrazený na Obr. 6.7.

Predstavme si, že máme k dispozícii merania všetkých stavov⁴ a napriek tomu používame pozorovateľ s dokonale presným modelom (merania sú nedokonalé). V takomto prípade možno rozdiel medzi skutočnou a odhadnutou hodnotou reprezentovať pomocou premennej chyby odhadu $z(t)$:

$$z(t) = x(t) - \hat{x}(t). \quad (6.20)$$

Na základe toho môžeme uvedenú rovnici prepísať do diferenciálnej podoby. Tá charakterizuje dynamiku chyby odhadu:

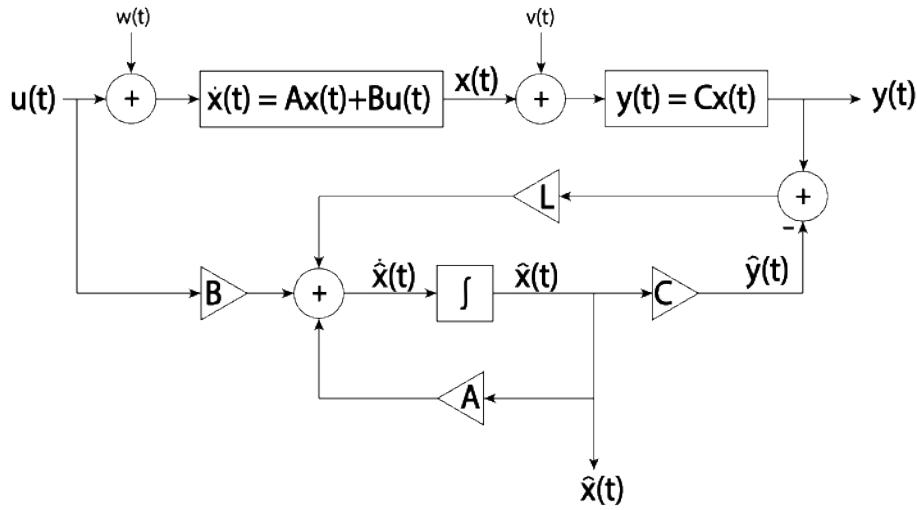
$$\dot{z}(t) = \dot{x}(t) - \dot{\hat{x}}(t), \quad (6.21)$$

$$\dot{z}(t) = Ax(t) + Bu(t) - (A\hat{x}(t) + Bu(t)) = A(x(t) - \hat{x}(t)), \quad (6.22)$$

$$\dot{z}(t) = Az(t). \quad (6.23)$$

Riešením tejto diferenciálnej rovnice je $z(t) = e^{At}z(0)$, kde $z(0)$ predstavuje počiatočný rozdiel medzi reálnym systémom a tým simulovaným. Z riešenia vyplýva, že takýto typ pozorovateľa by časom konvergoval ku skutočnej hodnote systému, teda $\lim_{t \rightarrow \infty} z(t) = 0$, jedine v prípade, že systém opísaný maticou A by bol stabilný (póly matice by boli záporné a teda e^{At} by sa s

⁴ $y(t) = x(t)$.



Obr. 6.8: Schéma Luenbergerovho pozorovateľa.

časom zmenšovalo). Ale úlohou pozorovateľa je zabezpečiť nájdenie skutočnej hodnoty stavov, bez ohľadu na stabilitu systému.

Riešenie tohto problému spočíva v pridaní späťnej väzby chyby odhadu, ktorá sa vracia cez vhodne zvolený násobiteľ [27, Part 2: State Observers]. Principiálne aplikujeme to, čo bolo uvedené v Kap. 6.1. Rozdiel je len v tom, že nevraciame stavy $x(t)$ cez konštantu K , ale ich rozdiel $z(t) = x(t) - \hat{x}(t)$ cez násobiteľ L . Vďaka tomu možno dynamiku chyby odhadu $z(t)$ vyjadriť ako

$$\begin{aligned}\dot{z}(t) &= \dot{x}(t) - \dot{\hat{x}}(t), \\ \dot{z}(t) &= Ax(t) + Bu(t) - (L(Cx(t) - C\hat{x}(t)) + A\hat{x}(t) + Bu(t)), \\ \dot{z}(t) &= (A - LC)(x(t) - \hat{x}(t)) = (A - LC)z(t).\end{aligned}\quad (6.24)$$

V takomto prípade je riešením $z(t) = e^{(A-LC)t}z(0)$ kde vlastné čísla matice $A - LC$ závisia od toho, ako zvolíme hodnoty L . Takže aj keď je systém nestabilný, pomocou násobiteľa L vieme stabilizovať odchýlku meraných a odhadnutých hodnôt. Systém si tak nestabilitu zachová, no napriek tomu budeme vedieť určiť hodnoty jeho stavov v čase. Takýto typ pozorovateľa sa nazýva *stavový pozorovateľ* alebo tiež *Luenbergerov pozorovateľ* a na Obr. 6.8 je zobrazená schéma tohto riešenia. Pre aplikáciu pozorovateľa v DT platia totožné pravidlá ako do teraz, teda transformácia matíc pomocou Rov. (6.12) a voľba pôlov na základe Rov. (6.9).

Takže *Luenbergerov pozorovateľ* na základe modelu sleduje nemerateľné stavy a vracia nám ich odhadované hodnoty. Podobný koncept je použitý aj pri ďalšom algoritme na odhad skutočnej hodnoty stavov nazývaného *Kalmanov filter*, ktorý však už berie ohľad aj na skutočnosť, že dátu môžu byť zašumené.

Hlavným rozdielom Kalmanovho filtra od opísaného stavového pozorovateľa je, že hodnota násobiaca rozdiel odhadnutej a meranej veličiny K_{kal} (Kalmanovo zosilnenie) už nie je konštantná⁵, ale je závislá od procesného a meracieho šumu $w(t)$ a $v(t)$. Keďže ale priebeh šumu v čase nepoznáme, jej opis je reprezentovaný takzvanými kovariančnými maticami procesného šumu Q a meracieho šumu R .

⁵Zvyčajné značenie v literatúre je K , ale vzhľadom na predošlé kapitoly by to mohlo pôsobiť mätúco.

Okrem tohto zásadného rozdielu sa Kalmanov filter využíva predovšetkým v DT. Nakoľko odhad stavu pozostáva z dvoch častí, reprezentácia pre DT je omnoho jednoduchšia⁶.

Prvá časť, tiež nazývaná apriórna alebo predikčná, slúži na odhad stavu \hat{x}_k^- na základe modelu a na výpočet kovariancie chyby P_k^- , kde mínusové znamienko značí prvotný odhad v n-tom kroku algoritmu:

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1}, \\ P_k^- &= AP_{k-1}A^T + Q.\end{aligned}\tag{6.25}$$

V druhej, aposteriórnej časti, sa upravuje prvotný odhad a spresňuje sa na základe merania. Taktiež sa tu počíta spomínaný faktor K_{kalk} :

$$\begin{aligned}K_{\text{kalk}} &= \frac{P_k^- C^T}{C P_k^- C^T + R}, \\ \hat{x}_k &= \hat{x}_k^- + K_{\text{kalk}}(y_k - C\hat{x}_k^-), \\ P_k &= (I - K_{\text{kalk}}C)P_k^-.\end{aligned}\tag{6.26}$$

Takže našou hlavnou rovnicou odhadujúcou stavy je

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_{k-1} + K_{\text{kalk}}(y_k - C(A\hat{x}_{k-1} + Bu_{k-1})).\tag{6.27}$$

Ostatné vzťahy sú súčasťou algoritmu z dôvodu výpočtu najoptimálnejšej hodnoty K_{kalk} , ktorého výpočet sme schopní upravovať pomocou matíc Q a R . Keďže návrh týchto matíc je úlohou používateľa, ukážem to, akým spôsobom výsledný odhad ovplyvňujú.

Ak sa pozrieme na prvú rovinu v (6.26), je to jediný vzťah v ktorom sa nachádza matica R . Uvažujme čo sa s Kalmanovým zosilnením stane ak $R \rightarrow 0$:

$$\lim_{R \rightarrow 0} K_{\text{kalk}} = \frac{P_k^- C^T}{C P_k^- C^T + 0} = \frac{P_k^- C^T}{C P_k^- C^T} = C^{-1}\tag{6.28}$$

a teda

$$\hat{x}_k = \hat{x}_k^- + C^{-1}(y_k - C\hat{x}_k^-) = C^{-1}y_k.\tag{6.29}$$

Tento výsledok môže pôsobiť nesprávne, nakoľko matica C nebýva zvyčajne invertovateľná a tým pádom ani $R \neq 0$. Ak by sme však uvažovali systém s jediným stavom, ktorý je zároveň meraným, tak matica $C = 1$ a v takom prípade $C^{-1} = 1$. Po tejto úvahе Rov. (6.29) hovorí toľko, že čím menšie R zvolíme, tým viac sa budeme spoliehať pri odhade na namerané dátu [27, Part 4: An Optimal State Estimator Algorithm].

Pri matici Q je pochopenie, ako ovplyvňuje odhad stavov trošku náročnejšie, nakoľko do vzťahu pre výpočet K_{kalk} je vnášaná prostredníctvom P_k . Ale uvažujme opäť veľmi jednoduchý systém s jediným stavom, ktorý zároveň aj meríame, teda $C = 1$. V Rov. (6.26) pre K_{kalk} bude čitateľ menší ako menovateľ (o hodnotu R) a teda $K_{\text{kalk}} < 1$. Na základe toho a poslednej rovnice v (6.26) môžeme povedať, aj že $P_k^- > P_k$. A teraz povedzme, že matica $A = 0.5$ (je potrebný stabilný systém a ten je v DT definovaný pôlmi $p < |1|$). V takomto prípade vidíme, že v Rov. (6.25) bude člen $AP_k^- A^T$ s iteráciami konvergovať k nule (myslím tým že dynamika člena má

⁶Zmena značenia DT: namiesto premennej $[k]$ pre vzorku signálu budem používať dolný index s premenou k . Takéto značenie je v tomto prípade prehľadnejšie.

charakter sa s iteráciami zmenšovať, hodnotu 0 však nedosiahne práve vďaka matici Q). Ak by sme zároveň zaviedli $Q = 0$, vývoj kalmanovho zosilnenia bude

$$\lim_{P_k \rightarrow 0} K_{\text{kalk}} = \frac{P_k^- C^T}{C P_k^- C^T + R} = \frac{0}{0 + R} = 0 \quad (6.30)$$

a tým pádom odhad stavov bude záležať len od hodnôt predikovaných modelom:

$$\hat{x}_k = \hat{x}_k^- + K_{\text{kalk}}(y_k - C\hat{x}_k^-) = \hat{x}_k^- \quad (6.31)$$

Inými slovami možno povedať, že čím menšie Q zvolíme, tým viac sa budeme spoliehať na odhad na základe modelu.

Matice R a Q v skutočnosti reprezentujú kovariančné matice šumu merania a procesného šumu. V tomto duchu možno teóriu chápať tak, že matice R má na hlavnej diagonále variáciu šumu merania pre daný senzor a mimo diagonály sú hodnoty vyjadrujúce koreláciu medzi jednotlivými meraniami. Tak isto aj matice Q , definuje variáciu šumu modelu opisujúceho daný dej, pričom na hlavnej diagonále je variácia jednotlivých stavov (vyjadrená v ich jednotkách) a mimo diagonály je vyjadrené, akým spôsobom šum jedného a druhého stavu spolu súvisia.

Takže malé hodnoty matice R vravia, že senzory sú nezašumené a presné. Hodnoty matice by mali byť, v závislosti od šumu, o dva a viac rádov menšie ako sú merania. Malé hodnoty matice Q hovoria, že model systému, ktorý sme navrhli je presný. Hodnoty matice by opäť mali dávať význam, avšak zistenie kovariancie pre systém je náročnejšie ako pre senzory, keďže systém zahrňa množstvo zjednodušení. Takže ladenie tejto matice môže viesť k neočakávaným hodnotám.

6.4 PID

Prvým riadením, ktoré bolo na MagnetoShiled implementované bol proporčne-integračno-derivačný regulátor, tiež známy ako PID regulátor. Ten som na MagnetoShield implementoval už pri prvotnom návrhu tohto zariadenia. Jedná sa o spätnovázobné riadenie, založené na vyhodnocovaní regulačnej odchýlky $z(t)$. Tá je výsledkom rozdielu sledovanej referencie $r(t)$ a regulovanej veličiny $y(t)$:

$$z(t) = r(t) - y(t). \quad (6.32)$$

Následne je tento rozdiel použitý pre vytvorenie vstupnej veličiny, ktorá je sumou troch zložiek spracúvajúcich tento signál.

Prvou zložkou je proporcionálna časť. Jedná sa o násobiteľ K_P násobiaci regulačnú odchýlku:

$$u(t) = K_P z(t). \quad (6.33)$$

Samotné proporčné riadenie je jednou z možností regulácie systému. V takomto prípade je vstup do systému závislý jedine na násobení regulačnej odchýlky. Toto riadenie ale vedie k fenoménu, tzv. trvalej regulačnej odchýlke. Jedná sa o ustálenie regulovanej veličiny na hodnote súčasťou referencií, ale nie s ňou totožnej. Je to spôsobené tým, že ak spojíme Rov. (6.32) s Rov. (6.33), referenciu možno vyjadriť ako

$$r(t) = y(t) + \frac{1}{K_P} u(t). \quad (6.34)$$

Z toho je zreteľne vidno, že referenciu sme schopní dosiahnuť len vtedy, ak je systém stabilizovateľný v žiadanej hodnote nulovým vstupom $u(t)$, čo pre väčšinu systémov (ako aj pre

MagnetoShield) neplatí.

Z tohto dôvodu sa k proporcionálnemu členu pridáva integračný člen s násobiteľom K_I . Ten je zodpovedný za integráciu (v DT sumáciu) odchýliek $z(t)$. Vstup tvorený týmto členom je daný ako

$$u(t) = K_I \int_0^t z(\tau) d\tau. \quad (6.35)$$

Vďaka tomu sme schopní odstrániť trvalú regulačnú odchýlku, avšak za cenu istej oscilácie okolo žiadanej hodnoty v dôsledku "vyrovnávania znamienok" v integráli. Taktiež, aby integrál nenabral príliš vysoké hodnoty, je prospešné jeho softvérové ohraničenie. Toto ošetrenie sa tiež nazýva "Anti-windup".

Posledným členom PID regulátora je derivačný člen s násobiteľom K_D . Časť vstupu generovaná týmto členom je definovaná vzťahom

$$u(t) = K_D \frac{dz(t)}{dt}. \quad (6.36)$$

Z uvedeného vyplýva, že prírastok spôsobený derivačným členom je závislý od toho, ako "strmo" sa riadená veličina blíži k referencií, čo v konečnom dôsledku skracuje čas dosiahnutia žiadanej hodnoty.

Výsledná forma PID regulátora v CT je

$$u(t) = K_P z(t) + K_I \int_0^t z(\tau) d\tau + K_D \frac{dz(t)}{dt} \quad (6.37)$$

a v DT je

$$u[k] = K_P z[k] + T_s K_I \sum_{i=0}^k z[i] + K_D \frac{z[k] - z[k-1]}{T_s}, \quad (6.38)$$

kde T_s predstavuje periódu vzorkovania.

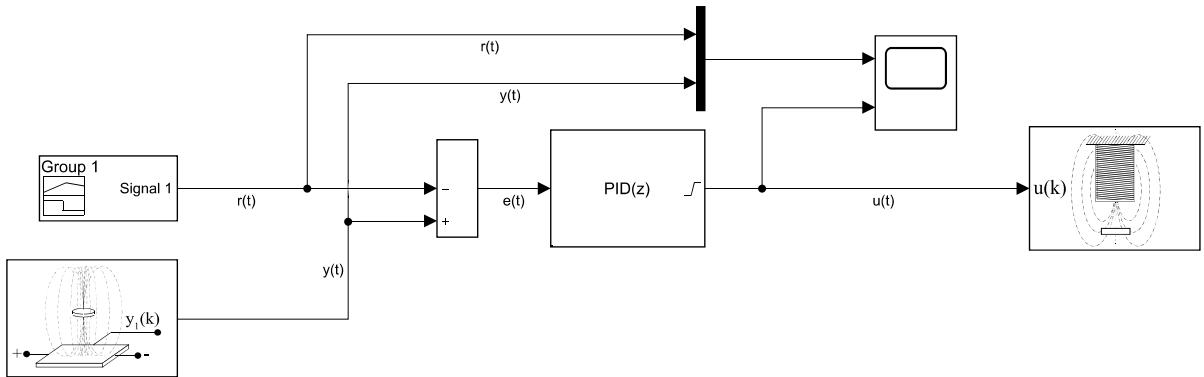
Pri implementácii PID riadenia na systém MagnetoShield v rámci rozhrania Simulink bol v bloku *PID* použitý tvar algoritmu v podobe

$$\begin{aligned} u[k] &= P(z[k] + T_s I \sum_{i=0}^t z[i] + D \frac{z[k] - z[k-1]}{T_s}), \\ P &= K_P, \\ I &= \frac{K_I}{K_P}, \\ D &= \frac{K_D}{K_P}. \end{aligned} \quad (6.39)$$

Vzhľadom na tento tvar som pri riadení PID regulátorom zvolil nasledovné ladenie uvedených konštánt:

$$P = 3.500 \quad I = 1.670 \quad D = 0.025$$

a nakoľko sa jedná o jednoduchý algoritmus tak som pri riadení Arduinom DUE mohol zvoliť rýchle vzorkovanie $T_s = 1$ ms.



Obr. 6.9: Schéma riadenia PID regulátorom.

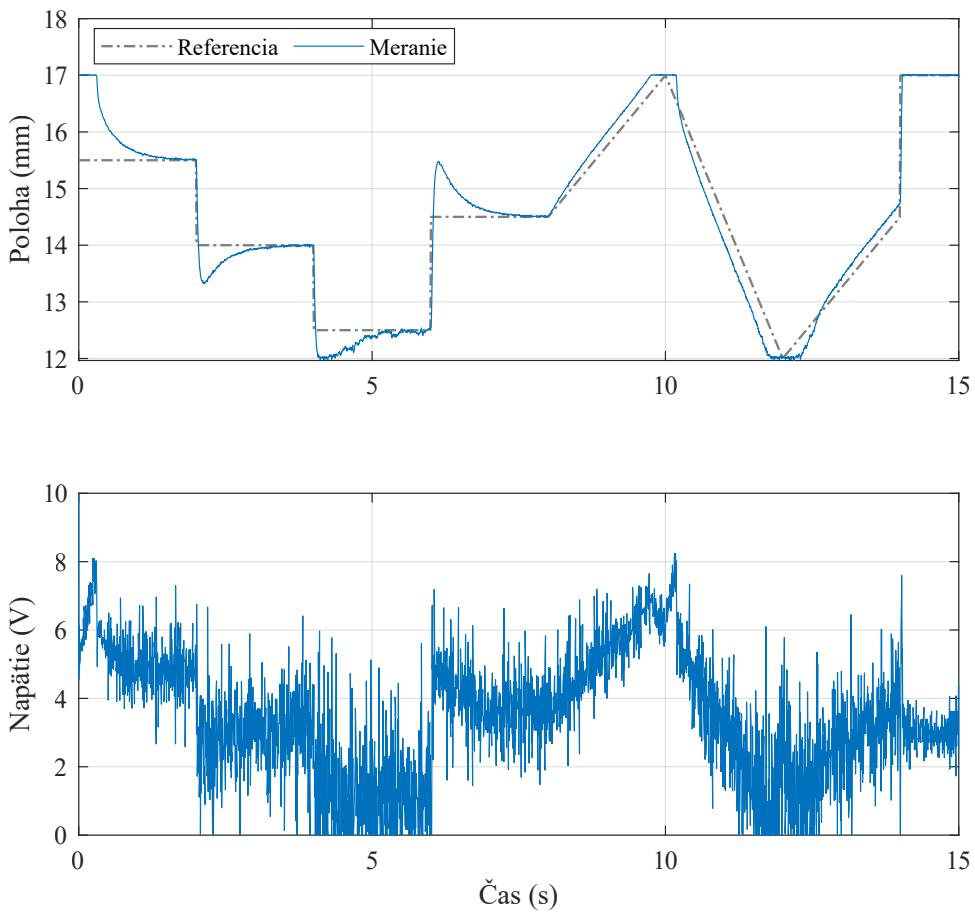
Okrem uvedeného tvaru existuje ešte jeden spôsob, akým sa dajú vyjadriť konštanty PID regulátora a tento tvar bol využitý pri tvorbe funkcií určených pre PID riadenie v rámci iniciatívy AutomationShield pri aplikačnom rozhraní Arduino IDE:

$$u[k] = K_P(z[k] + \frac{T_s}{T_i} \sum_{i=0}^t z[i] + T_d \frac{z[k] - z[k-1]}{T_s}),$$

$$T_i = \frac{K_P}{K_I},$$

$$T_d = \frac{K_D}{K_P}.$$
(6.40)

Pri pohľade na schému riadiaceho programu v Simulinku na Obr. 6.9 je potrebné upozorniť na jeden zásadný fakt. V schéme blok na výpočet regulačnej odchýlky $z[k]$ má prehodené znamienka pri porovnaní s Rov. 6.32. Je to z dôvodu, ako je nás systém navrhnutý. Intuícia u väčšiny systémov napovedá, že kladným vstupom dostaneme pri riadenej veličine kladnú zmenu. Ak by riadenou veličinou bola výška levitácie, tak by bola táto úvaha v poriadku. V našom prípade ale nemeríme výšku levitácie, ale vzájomnú vzdialenosť medzi elektromagnetom a permanentným magnetom. Ich vzájomná vzdialenosť sa práve naopak, rastúcim vstupom v kladnom smere zmenšuje. Výsledkom opísanej logiky je priebeh riadenia, ktorý je zobrazený na Obr. 6.10.



Obr. 6.10: Priebeh regulovanej veličiny pri riadení PID regulátorom.

6.5 Riadenie uložením pólov

Riadenie uložením pólov je asi najpriamočiarejšia implementácia teórie opísanej v Kap. 6.1, 6.2 a 6.3. Hlavným cieľom pri návrhu regulátora bude, ako názov algoritmu napovedá, voľba konštanty K tak, aby sa nám podarilo uložiť póly matice $(A - BK)$ z Rov. (6.11) do stabilnej oblasti.

Model s ktorým budeme pracovať je odvodený z Rov. (5.42) a po jeho diskretizácii Rov. (6.12) pre vzorkovaciu períodu $T_s = 0.0015$ s dostávame rovnicu dynamiky:

$$\begin{bmatrix} \Delta x_1[k+1] \\ \Delta x_2[k+1] \\ \Delta x_3[k+1] \end{bmatrix} = \begin{bmatrix} 1.0011 & 0.0015 & -0.0001 \\ 1.4923 & 0.9987 & -0.1661 \\ 0.0147 & 0.0173 & 0.4345 \end{bmatrix} \begin{bmatrix} \Delta x_1[k] \\ \Delta x_2[k] \\ \Delta x_3[k] \end{bmatrix} + \begin{bmatrix} 0 \\ -0.0004 \\ 0.0026 \end{bmatrix} \Delta u[k]. \quad (6.41)$$

Následne aplikáciou Rov. (6.17) získame rozšírenú formu stavového priestoru, umožňujúcu sledo-

vanie zadanej referencie $r[k]$:

$$\begin{bmatrix} \Delta x_1[k+1] \\ \Delta x_2[k+1] \\ \Delta x_3[k+1] \\ x_i[k+1] \end{bmatrix} = \begin{bmatrix} 1.0011 & 0.0015 & -0.0001 & 0 \\ 1.4923 & 0.9987 & -0.1661 & 0 \\ 0.0147 & 0.0173 & 0.4345 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1[k] \\ \Delta x_2[k] \\ \Delta x_3[k] \\ x_i[k] \end{bmatrix} + \begin{bmatrix} 0 \\ -0.0004 \\ 0.0026 \\ 0 \end{bmatrix} \Delta u[k] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} r[k]. \quad (6.42)$$

Na to, aby sme mohli pomocou riadiaceho zákona získať podobu reprezentácie, ako má Rov. (6.16), treba zvážiť, ktoré póly sú nestabilné a navrhnutú ich novú pozíciu. O stabilité stavov získame prehľad nájdením pôlov systému (6.42), označených ako α_o ⁷. Hodnoty pôlov systému MagnetoShield sú

$$\alpha_o = \begin{bmatrix} 1.0297 \\ 0.9662 \\ 0.5782 \\ 1 \end{bmatrix}. \quad (6.43)$$

Tu vidíme, že poloha je nestabilný stav, rýchlosť je blízko pri okraji jednotkovej kružnice, takže stabilita sa dosahuje pomerne pomaly no elektrický prúd je veľmi stabilný. Možno to odôvodniť tým, že prúd je veľmi úzko spätý s napäťom a ostatné veličiny majú na jeho zmenu minimálny vplyv, pričom vývoj polohy a rýchlosťi v čase sú omnoho komplexnejšie. Posledný pól sa nachádza na hranici stability a charakterizuje chovanie sa veličiny x_i . Tu ale vzniká problém, a to ten že tento umelo vytvorený stav nie je pozorovateľný (viď Rov. (6.46)), preto má aj vlastné číslo hodnotu 1 pre DT (v CT by tento pól vyšiel s hodnotou 0, takže opäť na hranici stability). To spôsobuje, že stabilitu tohto stavu nie sme schopní takýmto spôsobom určiť. Naštastie pozorovateľnosť (možnosť odhadovať stav) a riaditeľnosť (možnosť ovplyvniť stav) nie sú identické a tento model ostáva riaditeľný (možno overiť pomocou Rov. (6.46)). Tým pádom môžeme navrhnuť novú hodnotu pôlov pre všetky stavy.

Nové hodnoty definujeme vektorom α_1 s tým, že ich umiestníme do stabilnej oblasti. Okrem toho všetky pôvodné póly systému ležia na reálnej osi, preto vyskúšajme dať novonavrhnuté póly s nenulovou imaginárnom zložkou:

$$\alpha_1 = \begin{bmatrix} 0.8591 \\ 0.7182 \\ 0.8953 + 0.4i \\ 0.8953 - 0.4i \end{bmatrix}. \quad (6.44)$$

Tieto póly sú očakávaným výsledkom pre systém, po aplikácii riadiaceho zákona, takže charakterizujú systém len v tvare, v akom je Rov. (6.16).

Na výpočet konštánt, ktorými budeme spätnú väzbu násobiť je nutné použiť niektorý z postupov na ich výpočet. Jedným takýmto postupom je tzv. *Ackermanov vzorec* [1]:

$$K = [0 \ \dots \ 0 \ 1] \varphi^{-1} \Phi(A), \quad (6.45)$$

⁷Póli sú vlastné čísla matice A , ale aby platila táto ekvivalencia pre všetky póly, systém musí byť pozorovateľný (angl. *observable*) [6, Chapter 27, Poles and Zeros of MIMO Systems], čo kvôli integračnému stavu x_i systém (6.42) nie je.

kde φ predstavuje maticu riaditeľnosti systému (angl. *controllability matrix*). Hodnosť tejto matice (počet lineárne nezávislých stĺpcov) by mal byť rovnaký ako rozmer matice A , ktorý je $n \times n$. Ak túto vlastnosť preveríme, zistíme, že uvedená reprezentácia v Rov. (6.42) nie je súčasťou pozorovateľnej ale je riaditeľnej. To znamená, že nie sme súčasťou schopných odhadnúť všetky stavy systému, ale sme ich určite schopní regulovať. Matica riaditeľnosti má následne tvar

$$\varphi = [B \ AB \ A^2B \ \dots \ A^{n-1}B]. \quad (6.46)$$

Matica $\Phi(A)$ reprezentuje charakteristickú rovnicu systému so žiadanými pôlmi. Pôly, ktoré si prajeme dosiahnuť α_1 , sú riešením tejto charakteristickej rovnice, pričom všeobecne má rovnica tvar

$$\Phi(s) = s^n + a_1s^{n-1} + a_2s^{n-2} + \dots + a_{n-1}s + a_n. \quad (6.47)$$

Koeficienty a_i , kde $i = 1, 2, 3, \dots, n$ sú práve také koeficienty, ktoré zabezpečujú, že riešením sú nami žiadane pôly α_1 . Pre výpočet a_i možno použiť v prostredí MATLAB funkciu `poly()`.

Akonáhle poznáme hodnotu a_i , funkcia $\Phi(s)$ je vyhodnotená pre $s = A$. Keďže náš systém má rozmer $n = 4$ (rozšírená forma stavového modelu o člen x_i), tak charakteristická rovinka vyhotovená pre systém opísaný maticou A bude

$$\Phi(A) = A^4 + a_1A^3 + a_2A^2 + \dots + a_3A + a_4. \quad (6.48)$$

Výsledkom výpočtu Rov. 6.45 je vektor K , násobiaci spätnú väzbu, v rozšírenom stavovom modeli (6.42):

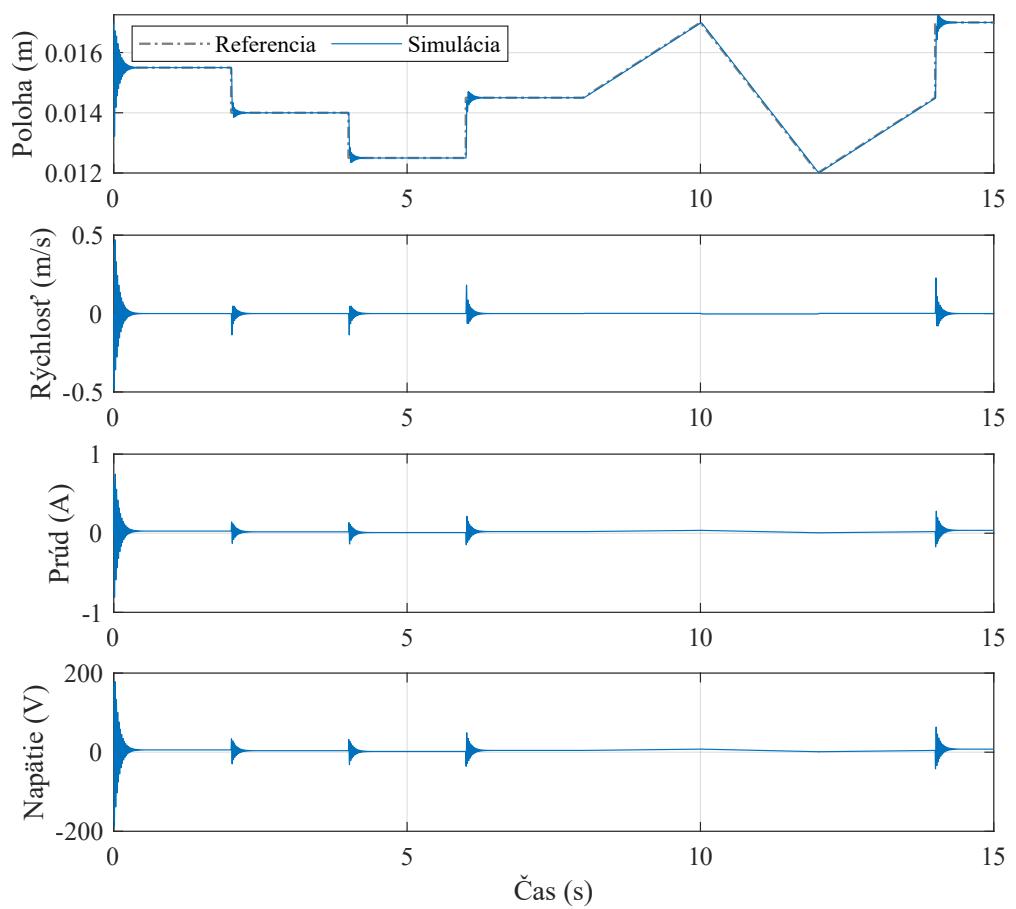
$$K = [-228449.78 \ -679.05 \ 36.34 \ 20963.14]. \quad (6.49)$$

Ako na Obr. 6.11 vidno, zvolené pôly naložaj systém stabilizujú a výška levitácie sleduje žiadanú hodnotu. Problém ale spočíva v tom, že ak sa pozrieme na priebehy prúdu a napätia, všimneme si, že ich hodnoty sú nad maximálnymi možnosťami reálneho systému. Je to spôsobené tým, že sme zvolili pôly príliš "hlboko" v stabilnej oblasti. Nakoľko každá akcia si vyžaduje energiu, takto vysoko postavené nároky na dosiahnutie stability jej potrebujú také množstvo, aké nie sme fyzicky schopní dodať. Preto je pri návrhu riadenia uložením pôlov nutné vziať do úvahy aj tento aspekt a tomu postavenie pôlov prispôsobiť. Taktiež sa objavil jav oscilácie. To je pre zmenu spôsobené nenulovou imaginárnu zložkou pôlov. Jej zmenšovaním by sa postupne oscilácia miernila a pri voľbe pôlu na reálnej osi úplne zanikne.

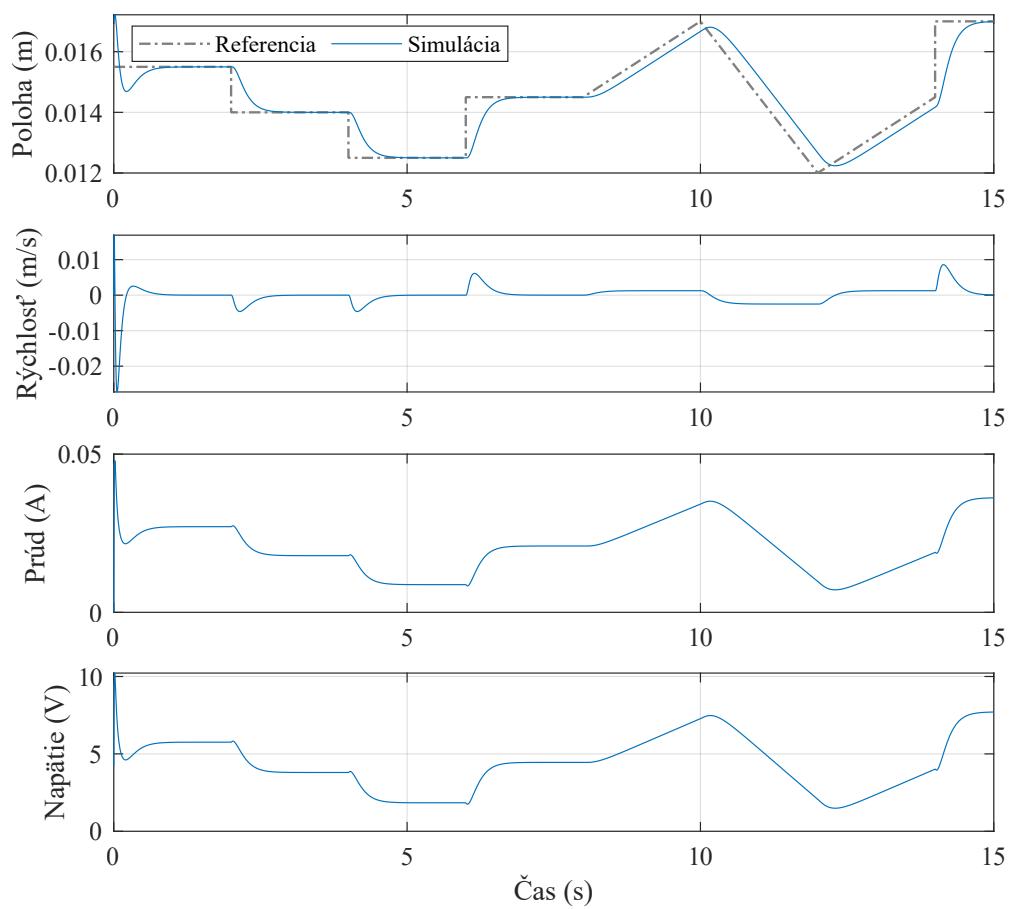
Opakoványmi pokusmi som došiel k záveru, že ideálne nastavenie pôlov a s ním súvisiace zosilnenie, ktoré sú vhodné aj pre reálny systém sú

$$\alpha_2 = \begin{bmatrix} 0.9891 \\ 0.9260 \\ 0.8353 \\ 0.9850 \end{bmatrix}, \quad K = [-1869.52 \ -53.56 \ 86.93 \ 6.15].$$

Ich postavenie je omnoho bližšie k hranici stability a teda aj energetické nároky na vstup budú nižšie. Postavenie pôlov je taktiež čisto len na reálnej osi, takže nepredpokladáme žiadne oscilovanie. Simulovaný priebeh stavov je zobrazený na Obr. 6.12. Okrem zmeny pozície pôlov bol do schémy pridaný aj blok ohraničujúci maximálny vstup do systému na 10 V. To ale v tomto prípade nehralo zásadnú úlohu pri stabilite systému (v prípade veľmi agresívneho riadenia, ako



Obr. 6.11: Priebeh stavov s umiestnenými pólmi α_1 .



Obr. 6.12: Priebeh stavov s umiestnenými pólmi α_2 .

tomu bolo pri prvotnom návrhu pólov, toto obmedzenie spôsobuje nestabilitu systému).

Nakoľko návrh násobiteľa L pre Luenbergerov pozorovateľ je zadaním veľmi podobný návrhu násobiteľa K , rozhodol som sa tieto dva teoretické koncepty aplikovať naraz. Pre jednoduchosť návrhu budeme uvažovať model, ktorý má meraný jediný stav a to polohu (môžeme si to dovoliť, nakoľko vieme že model má pozorovateľné všetky stavy okrem x_i). Merania prúdu a rýchlosť budú založené na odhade⁸.

Z Kap. 6.3, Rov. (6.24) je zrejmé, že voľba násobiteľa L má podobný význam, ako má zosilnenie K a teda stabilizáciu pólov (tomto prípade stabilizujeme chybu medzi výstupom z pozorovateľa a meranou veličinou $\dot{z}(t) = (A - LC)z(t)$, alebo v diskrétnom tvare $z[k+1] = (A_d - LC_d)z[k]$). Vďaka takejto analógií môžeme predpokladať, že na výpočet hodnôt L , na základe navrhnutých pólov, možno použiť taktiež Ackermanov vzorec. Vzťah sa ale jemne pozmení, nakoľko nie je pre nás v tomto momente podstatná riaditeľnosť ale pozorovateľnosť (angl. *observability*) [6, Chapter 29, Observers, Model-based Controllers]. Taktiež neberieme do úvahy stav x_i , takže ne- uvažujeme rozšírenú formu stavového priestoru, ale budeme pracovať s Rov. (6.41). V takomto prípade Ackermanov vzorec pre výpočet L má podobu

$$L = \Phi(A)\vartheta^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (6.50)$$

$\Phi(A)$ má v tomto prípade rovnaký význam ako vyššie a teda, je to charakteristická rovnica pre nami zvolené póly p vyhodnotená pre $s = A$. ϑ predstavuje maticu pozorovateľnosti systému a jej hodnosť by mala byť opäť n (v prípade, že to je menej, nie sme schopní odhadnúť všetky stavy):

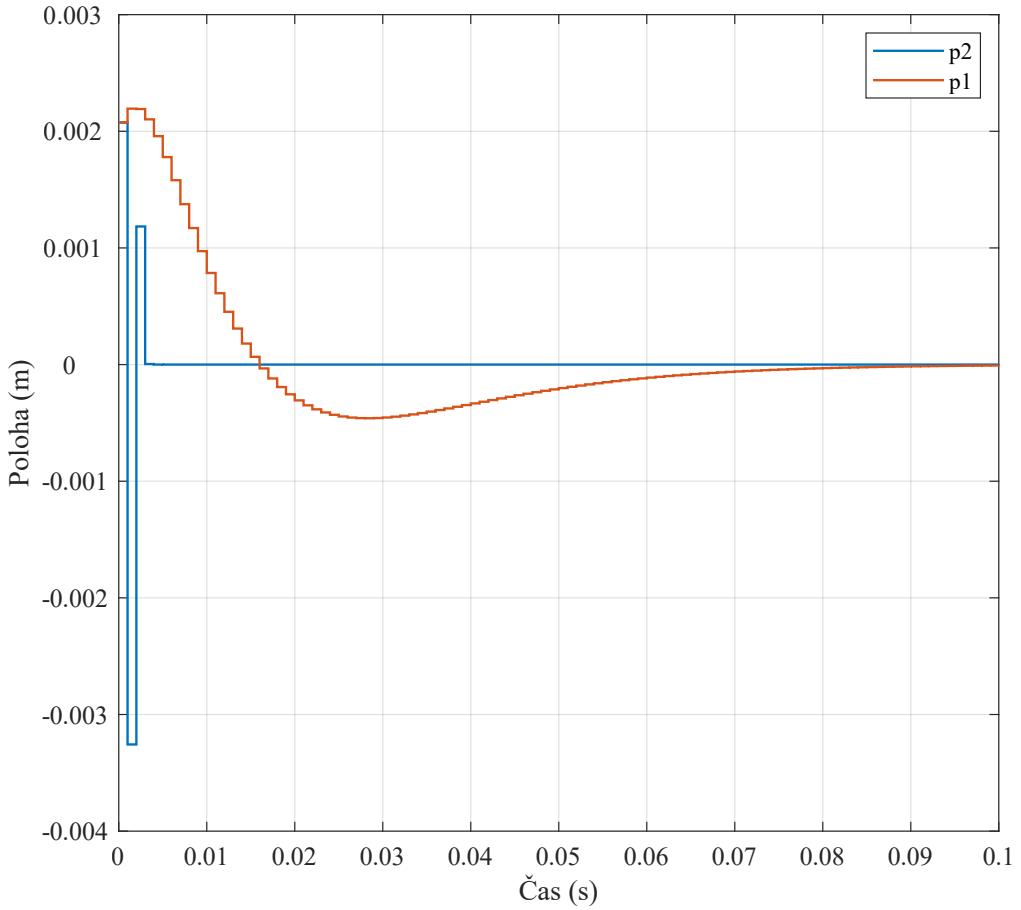
$$\vartheta = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}. \quad (6.51)$$

Pri návrhu pólov pre pozorovateľ máme jednu zásadnú výhodu a to tú, že odhad nie je závislý na vstupe. Tým pádom nemusíme brať ohľad na energetické požiadavky rýchlych pólov - môžeme ich urobiť tak rýchle ako potrebujeme. Pre porovnanie, aký dopad má voľba pólov na odhad stavov, je na Obr. 6.13 zobrazený vývoj chyby odhadu polohy $z[k]$ pre póly p_1 a pre p_2 (v systéme nie je uvažovaný šum a počiatočná chyba je pre oba systémy rovnaká). Síce polohu meríame, ale pre ukážku to je ideálny stav, nakoľko jeho dynamika je pomalšia ako pre prúd a ľahšie interpretovateľná ako rýchlosť. Analogicky ale možno očakávať rovnaký vývoj chyby aj pri ostatných stavoch. Voľba pólov p_1 a p_2 bola následovná:

$$p_1 = \begin{bmatrix} 0.93 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad p_2 = \begin{bmatrix} 0.003 \\ 0.0006715 \\ 0.00003715 \end{bmatrix}.$$

Na Obr. 6.13 oba priebehy konvergujú k meranej hodnote – teda chyba je nulová, ale rozdiel je, že čím dlhšie budeme mať nepresné odhady stavov, tým dlhšie sa bude regulátor chovať nesprávne, čo v konečnom dôsledku už stabilitu systému ovplyvniť môže (síce pozorovateľ v prípade p_1

⁸Keďže rýchlosť sa dá aproximovať jednoduchou deriváciou, na prúde bude lepšie vidno rozdiel medzi odhadom Luenbergerom a neskôr Kalmanom.



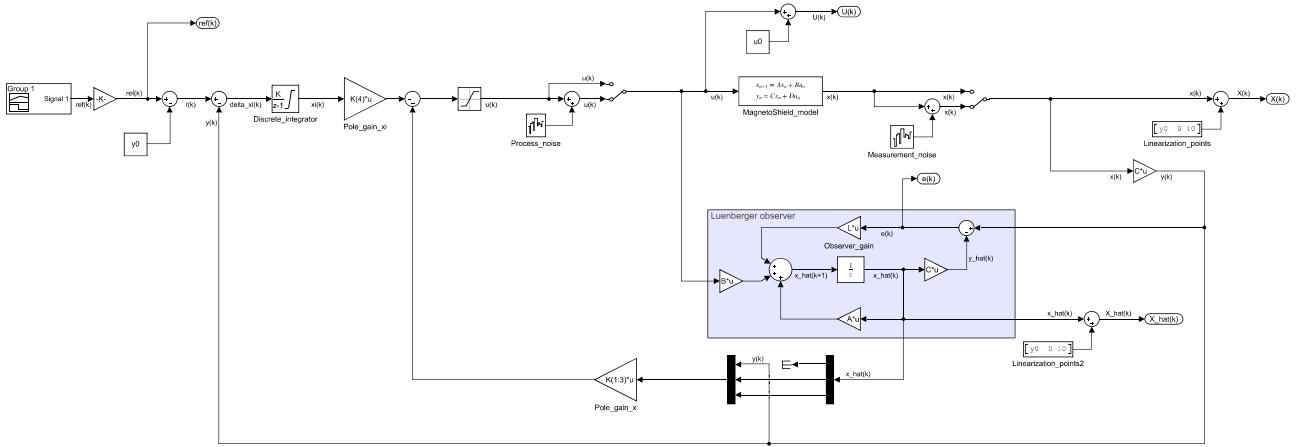
Obr. 6.13: Vplyv voľby pólov na dynamiku pozorovateľa.

konverguje behom 0.1 s, ale treba pamätať, že vzorkovanie je $T_s = 0.0015$ s, čo predstavuje 67 nesprávnych vstupov). Preto som pre implementáciu do skutočného systému zvolil póly p_2 , ktorým odpovedá zosilnenie

$$L = \begin{bmatrix} 2.43 \\ 1032.82 \\ -464.23 \end{bmatrix}. \quad (6.52)$$

Výsledkom tohto procesu je riadenie pomocou uloženia pólov. Pred jeho záverečnou implementáciou do systému som vytvoril jeho simuláciu, ktorá testuje kooperáciu navrhnutého faktora K so zosilnením L , či rýchlosť konvergencie chyby je dostatočne rýchla na to, aby sme boli schopní systém uriadiť. V tomto modeli je už aj zakomponovaný šum, nakoľko to má byť čo najvernejšia napodobenina reálneho systému. Schéma simulácie je vidieť na Obr. 6.14 a výsledný vývoj stavových veličín, ako aj požadovanej referencie pre polohu a vývoj vstupnej veličiny vidno na Obr. 6.15.

Po implementácii navrhnutého riadenia na reálny systém sme dosiahli ustálenú levitáciu reprezentovanú Obr. 6.16. Riadiaci program je zobrazený na Obr. 6.17. Tu si v rámci schémy možno všimnúť jeden zásadný rozdiel a to, že je nutné vziať do úvahy linearizačné body x_{01}, x_{02}, x_{03} v

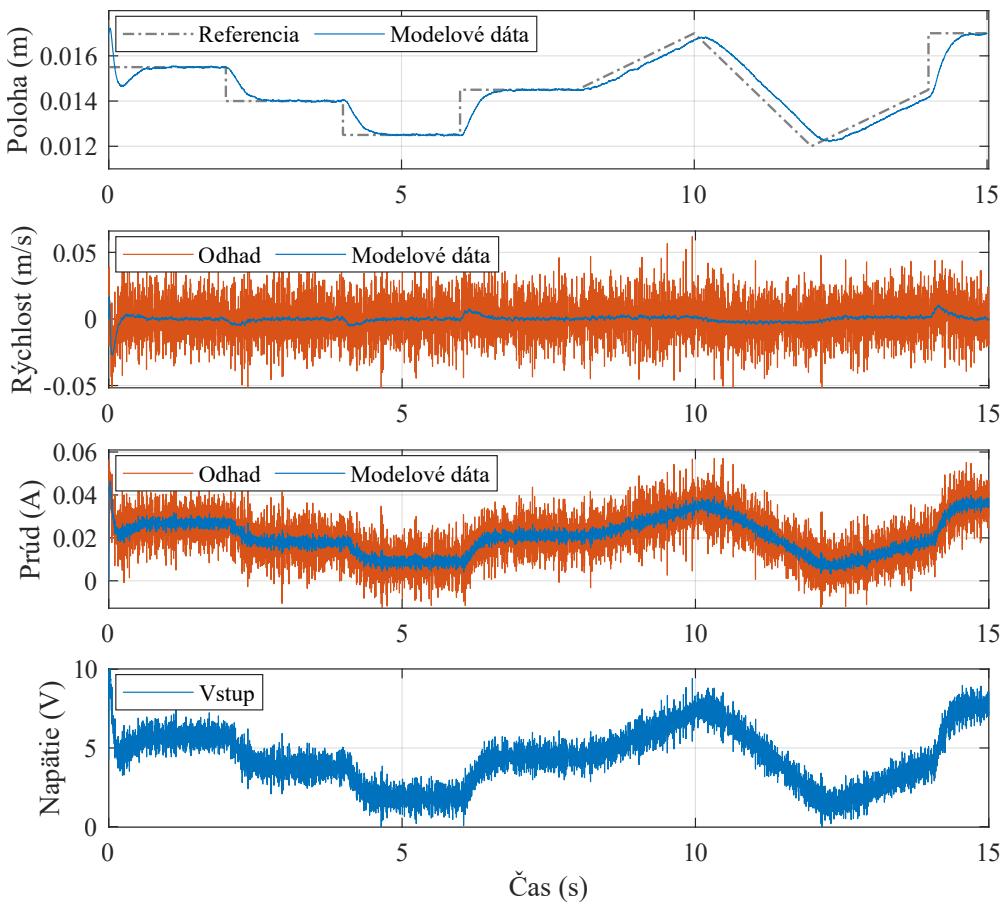


Obr. 6.14: Schéma simulačného programu pre komplexné testovanie funkčnosti navrhnutého riadenia uložením pólov.

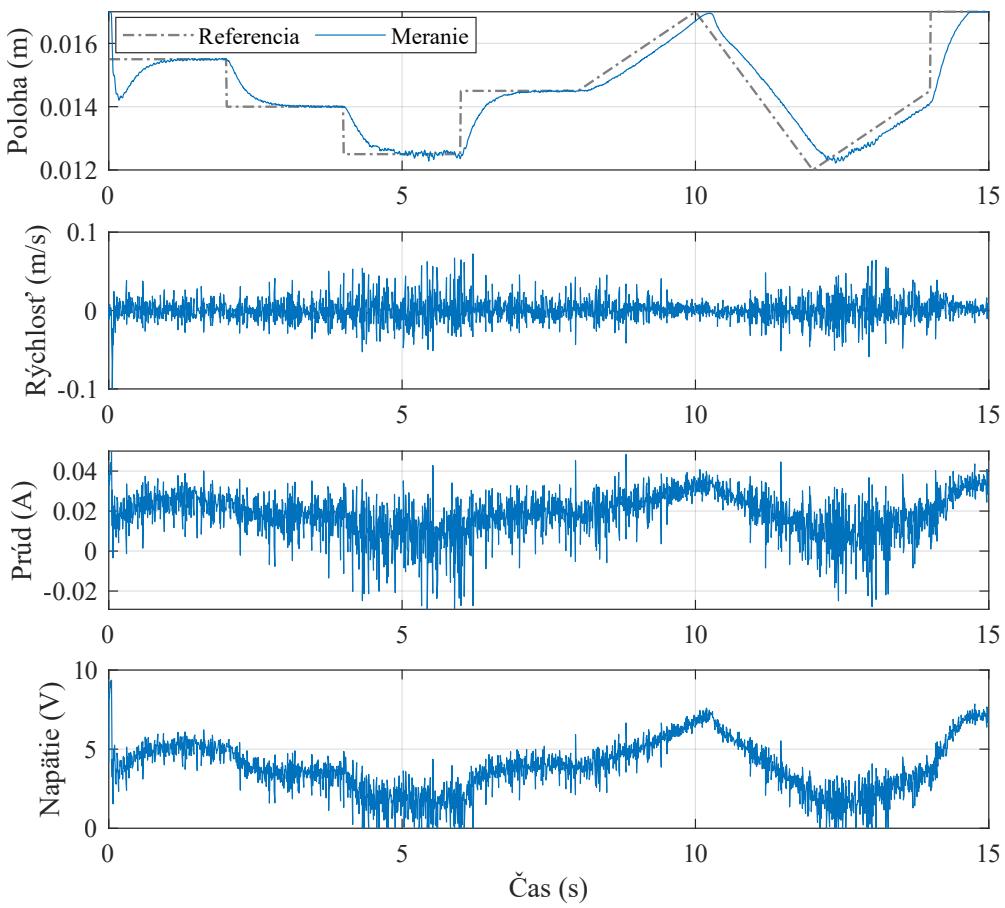
schéme reprezentované ako $y_0, 0, i_0$, nakoľko reálny systém nepracuje s relatívnymi premennými $\Delta x[k]$ ako model, ale s absolútными hodnotami $x[k]$.

Pri porovnaní priebehu so simuláciou je vidieť, že priebeh stavov je veľmi podobný simulovaným priebehom. Badateľné sú ale aj isté rozdielnosti, predovšetkým vo vývoji prúdu, ale nakoľko sa jedná o odhadované hodnoty a prúd v tomto prípade nie je riadená veličina, jeho výchylky možno zanedbať.

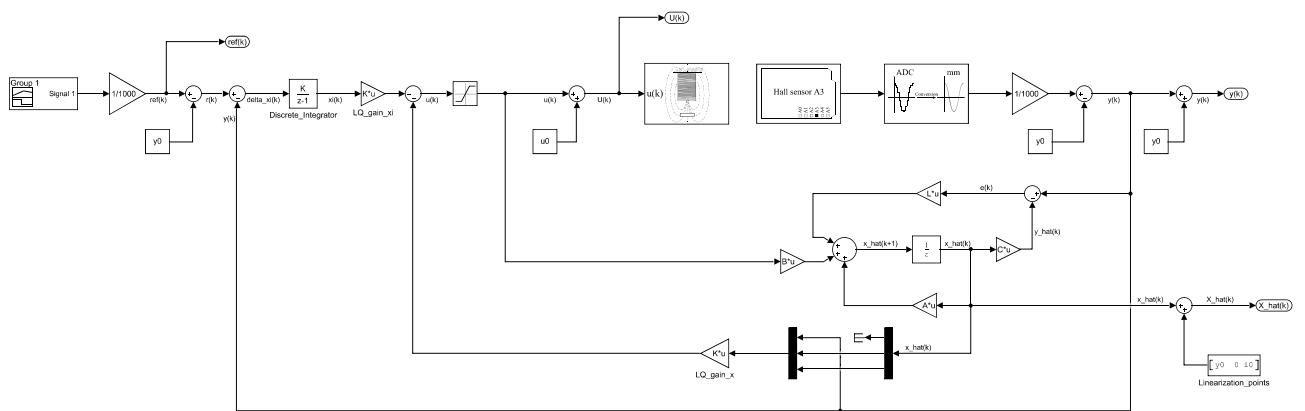
Asi najpatrnejším rozdielom je hustota dát. Je zrejmé, že experiment na reálnom systéme má redšiu hustotu zaznamenaných dát ako simulácia. Prirodzene sa dá uvažovať, či vzorkovanie $T_s = 0.0015$ s nie je pre Arduino DUE príliš rýchle. Tento efekt je však spôsobený prenosom medzi Arduinom a Simulinkom. Simulink dáva zaznamenávaniu dát nižšiu dôležitosť a aby bolo dodržané vzorkovanie riadiaceho programu, posielanie dát do Simulinku je občas prerušené. Tým pádom sú výsledné dáta skutočne redšie vzorkované. Dokonca toto vzorkovanie nemá pravidelnú periódu. Tento jav napriek tomu nemá dosah na dodržanie vzorkovania pre riadenie. Dôvodom použitia zobrazenia dát Simulinkom je to, že nám umožňuje vidieť kvalitu riadenia bezprostredne počas simulácie a pre tento účel je toto zobrazenie, aj napriek spomenutému nedostatku, postačujúce. Pre zaznamenávanie dát a pre ich následnú analýzu odporúčam využiť jeden zo spôsobov uvedených na konci Kap. 4.2 resp. využitie iného aplikáčného rozhrania (napr. Arduino IDE).



Obr. 6.15: Simulácia riadenia umiestnením pólov s použitím Luenbergerovho pozorovateľa uvažujúca prítomnosť šumu. „Modelové dátá“ reprezentujú simulovaný signál z modelu so zakomponovaným šumom, „Odhad“ je výstup z pozorovateľa.



Obr. 6.16: Priebeh riadenia umiestnenim pôlov s pouzitím Luenbergerovho pozorovateľa.



Obr. 6.17: Schéma riadenia uložením pôlov.

6.6 LQ regulátor

Lineárno-kvadratický regulátor (ďalej len *LQ regulátor* alebo *LQR*) je typ optimálneho riadiaceho algoritmu založeného na opise systému lineárnymi diferenciálnymi rovnicami, využívajúceho na optimalizáciu kvadratické programovanie. Pri porovnaní tohto typu riadenia oproti tomu, ktoré som opísal v predošej kapitole, dôjdeme k záveru, že oba prístupy sú v spôsobe, akým ovplyvňujú systém totožné. Opäť sa jedná o manipuláciu dynamiky systému, pomocou spätnovázobného zo-silnenia K , násobiaceho stavu systému. Rovnako tu platí aj princíp rozšírenia systému o stav opisujúci sledovanie referencie, ako to bolo opísané v Kap. 6.2 a následne využité v Rov. (6.42). Taktiež by bolo možné aplikovať Luenbergerov pozorovateľ a tým pádom aj využiť totožné riadiace programy pre Simulink, ktoré boli v minulej kapitole. Rozdielnosť oproti minulému riadeniu ale spočíva v optimalizačnej úlohe, ktorú sa v najbližších riadkoch posnažím priblížiť a nakoľko som implementáciu Luenbergerovho pozorovateľa uviedol, v tomto prípade využijem na odhad stavov Kalmanov filter opísaný na konci Kap. 6.3.

Riadenie uložením pólov má pomerne závažnú nevýhodu a to, že je veľmi neintuitívne. Ak máme na regulátor špecifické požiadavky ako napríklad, aby uregulovala veličiny v čo najkratšom čase, alebo ešte lepšie s použitím čo najmenšej energie, je veľmi náročné nájsť také póly, ktoré by túto požiadavku spĺňali. Dokonca, aby sme v rámci možností systému našli tie najlepšie póly zaistujúce požadované chovanie systému – optimálne póly, je pri predošom postupe otázkou štastia.

Za týmto účelom bola k návrhu spätnovázobného regulátora pridaná optimalizačná úloha. Cieľom tejto úlohy je nájsť také argumenty účelovej funkcie (tiež kriteriálna alebo objektová funkcia), aby minimalizovali prípadne maximalizovali jej hodnotu a pritom splňali podmienky, ak sú nejaké dané. Túto úlohu možno jednoducho definovať, ako hľadanie extrémov účelovej funkcie vrátane hraníc definovaných podmienkami. Správny návrh účelovej funkcie zabezpečuje, že nájdené argumenty minimalizujúce funkciu, sú v súlade, alebo v blízkosti ideálneho, nami očakávaného chovania sa regulátora.

Účelová funkcia, označme ju $J(\vec{x})$, je funkciou viacerých premenných, v tomto prípade charakterizovaných vektorom \vec{x} . Taktiež môžu byť pre ňu definované podmienky. A to podmienky rovnosti $h_i(\vec{x}) = 0$ a nerovnosti $g_j(\vec{x}) \leq 0$, kde i a j charakterizujú ich počet. Následne, zadanie úlohy má formu

$$\begin{aligned} & \min_x J(\vec{x}), \\ & h_i(\vec{x}) = 0, \\ & g_j(\vec{x}) \leq 0. \end{aligned} \tag{6.53}$$

pričom je dobré poznamenať, že väčšina optimalizačných úloh je formulovaná v podobe hľadania minima a aj maximalizácia sa dá preformulovať na minimizačnú úlohu

$$\max_x J(x) \leftrightarrow -(\min_x -J(x)). \tag{6.54}$$

Čo sa týka obmedzení optimalizačnej úlohy, budeme ďalej uvažovať jedine obmedzenia rovnosti, nakoľko nerovnosti nie je možné zahrnúť do pevného riadiaceho zákona, ktorý je našim konečným cieľom⁹ (obmedzenia nerovnosti budú priblížené v Kap. 6.7).

⁹Riadiacim zákonom je závislosť vstupu na spätnej väzbe stavov v tvare $\vec{u}_k = -K\vec{x}_k$

V prípade obmedzení rovnosti, musí pre extrémy funkcie platiť,

$$\begin{aligned}\nabla J(\vec{x}^*) &= \sum_{i=1}^m \lambda_i (\nabla h_i(\vec{x}^*)), \\ h_i(\vec{x}^*) &= 0,\end{aligned}\tag{6.55}$$

kde λ_i je konštantá. Táto formulácia vyjadruje skutočnosť, že v bode účelovej funkcie, ktorý je definovaný v rámci obmedzenia alebo obmedzení (ak je bod na rozhraní obmedzení, $\nabla h(\vec{x}^*)$ predstavuje vektor obsahujúci gradienty daných obmedzujúcich funkcií), sa nachádza extrém len vtedy, ak gradient účelovej funkcie v danom bode je kolineárny s gradientom obmedzenia. Ináč povedané, gradient funkcie $J(\vec{x}^*)$ a lineárna kombinácia gradientov funkcií $h_i(\vec{x}^*)$ ležia na tej istej priamke (pre viacej informácií vid' [16, Kap. 2.5]).

Vyjadrené podmienky pre extrém funkcie možno zobraziť pomocou jedinej účelovej funkcie nazývanej Lagrangeova funkcia, prípadne rozšírená účelová funkcia $\tilde{J}(\vec{x}, \vec{\lambda})$, ktorá má tvar:

$$\tilde{J}(\vec{x}, \vec{\lambda}) = J(\vec{x}) + \sum_{i=1}^m \lambda_i h_i(\vec{x}).\tag{6.56}$$

Už uvedená konštanta λ_i resp. vektor konštánt $\vec{\lambda}$ (obsahujúci všetky λ_i) sa nazýva Lagrangeov multiplikátor a jeho význam je zrejmí z Rov. (6.55), a teda násobí gradienty obmedzení tak, aby splňali rovnosť. Následne môžeme podmienku z Rov. (6.55), pomocou rozšírenej účelovej funkcie, zjednodušiť do zápisu

$$\nabla \tilde{J}(\vec{x}^*, \vec{\lambda}^*) = 0.\tag{6.57}$$

Táto rovnosť sa tiež nazýva "Nevyhnutná podmienka optimality prvého rádu" (ang. *First-order necessary condition of optimality*).

Teraz aplikujme podmienku optimality na úlohu pracujúcu so stavovým systémom. V prípade riešenia optimalizačnej úlohy pre stavový systém so stavmi \vec{x} a vstupmi \vec{u} , môžeme účelovej funkcií priradiť všeobecný tvar¹⁰:

$$J(\vec{x}, \vec{u}) = \Phi(\vec{x}_N, N) + \sum_{k=0}^{N-1} L_k(\vec{x}_k, \vec{u}_k).\tag{6.58}$$

Rád by som tu upozornil na zásadný fakt a to ten, že účelovú funkciu nepočítame v jedinom okamihu, ale jej hodnota závisí od vývoja premenných \vec{x} a \vec{u} počas sekvencie diskrétnych okamihov charakterizovaných pomocou k . Ich počet je N , kde N -tý okamih sa nazýva horizont. Potom účelová funkcia pozostáva z dvoch častí. $\Phi(\vec{x}_N, N)$ reprezentuje, ako veľmi hodnota posledného člena sekvencie na horizonte, teda člen \vec{x}_N , zaváži pri výpočte účelovej funkcie¹¹. Druhá časť $\sum_{k=0}^{N-1} L_k(\vec{x}_k, \vec{u}_k)$, nazývaná aj bežiaca suma (angl. *running cost*), charakterizuje prírastok ku účelovej funkcií počas doby, kym neprídeme k horizontu.

Obmedzenia reprezentujúce stavový systém možno zapísť ako

$$\vec{x}_{k+1} = \vec{h}_k(\vec{x}_k, \vec{u}_k),\tag{6.59}$$

¹⁰Opäť je použité značenie časových okamihov v podobe dolného indexu nakoľko to je v prípade tejto problematiky prehľadnejšie.

¹¹Všimnime si, Φ nie je funkciou \vec{u}_N . To je dané tým, že sa jedná o stavový model, kde \vec{u}_{k-1} zapríčinuje \vec{x}_k . Takže \vec{u}_N by ovplyvňovalo stav \vec{x}_{N+1} za horizontom.

resp.

$$\vec{h}_k(\vec{x}_k, \vec{u}_k) - \vec{x}_{k+1} = 0. \quad (6.60)$$

Tu \vec{h}_k reprezentuje vektor hodnôt, ktoré sú výsledkom jednotlivých rovníc stavového systému $h_i(\vec{x})$ (pre porovnanie $L_k(\vec{x}_k, \vec{u}_k)$ je tiež funkciou vektorov, ale výsledok je skalár).

Teraz môžeme sformulovať úlohu minimalizácie pre stavový systém ako

$$\begin{aligned} & \min J(\vec{x}_k, \vec{u}_k), \\ & \vec{h}_k(\vec{x}_k, \vec{u}_k) - \vec{x}_{k+1} = 0, \quad \vec{x}_0 = \text{známe} \end{aligned} \quad (6.61)$$

a následne použitím rovníc (6.56) a (6.58) uviesť rozšírenú účelovú funkciu

$$\tilde{J}(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) = \Phi(\vec{x}_N, N) + \sum_{k=0}^{N-1} L_k(\vec{x}_k, \vec{u}_k) + \sum_{k=0}^{N-1} \vec{\lambda}_k^T [\vec{h}_k(\vec{x}, \vec{u}) - \vec{x}_{k+1}]. \quad (6.62)$$

Tu si možno všimnúť, že už aj časť zahrňujúca obmedzenia je v sumácií, nakoľko obmedzenia musia platiť v každom diskrétnom okamihu až po horizont.

Pre zjednodušenie zápisu zjednotíme pod spoločnú sumu členy, charakteristické pre určitý diskrétny okamih substitúciou

$$H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) = L_k(\vec{x}_k, \vec{u}_k) + \vec{\lambda}_k^T h_k(\vec{x}, \vec{u}). \quad (6.63)$$

Funkcia $H(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k)$ sa nazýva "Hemiltonián" [16, Kap. 5.2] a s jeho použitím budeme hľadať gradient účelovej funkcie, ktorá má podobu:

$$\tilde{J}(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) = \Phi(\vec{x}_N, N) + \sum_{k=0}^{N-1} [H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) - \vec{\lambda}_k \vec{x}_{k+1}]. \quad (6.64)$$

Tu by som upozornil na ešte jednu úpravu tvaru Rov. (6.64) a to tú, že z prvého člena sumácie vyjmeme $H_0(\vec{x}_0, \vec{u}_0, \vec{\lambda}_0)$ a z posledného člena vyjmeme. $-\vec{\lambda}_{N-1} \vec{x}_N$.

$$\tilde{J}(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) = \Phi(\vec{x}_N, N) - \vec{\lambda}_{N-1} \vec{x}_N + H_0(\vec{x}_0, \vec{u}_0, \vec{\lambda}_0) + \sum_{k=0}^{N-1} [H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k)] \quad (6.65)$$

Ďalej bude výhodné, ak miesto gradientov priamo, vyjadríme najprv diferenciál rozšírenej účelovej funkcie $d\tilde{J}$ [4]. Celkový diferenciál si môžeme rozdeliť tak, že samostatne vytvoríme diferenciály funkcie s rešpektom pre rozdielne premenné. Tým pádom hľadáme taký bod, pre ktorý platí,

$$d\tilde{J}(\vec{x}^*, \vec{u}^*, \vec{\lambda}^*) = 0. \quad (6.66)$$

Vďaka tomuto tvaru môžeme diferenciál prvého a posledného stavu robiť zvlášť a vzhľadom na Rov. (6.66) dostávame podmienku optimality v tvare:

$$\begin{aligned} d\tilde{J}(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) &= (\nabla_{x_N} \Phi(\vec{x}_N, N) - \vec{\lambda}_{N-1})^T d\vec{x}_N + (\nabla_{x_0} H_0(\vec{x}_0, \vec{u}_0, \vec{\lambda}_0))^T d\vec{x}_0 \\ &+ \sum_{k=1}^{N-1} (\nabla_{x_k} H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) - \vec{\lambda}_{k-1})^T d\vec{x}_k + \sum_{k=0}^{N-1} (\nabla_{u_k} H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k))^T d\vec{u}_k \\ &+ \sum_{k=0}^{N-1} (\nabla_{\lambda_k} H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) - \vec{x}_{k+1})^T d\vec{\lambda}_k = 0. \end{aligned} \quad (6.67)$$

Keďže nutná podmienka optimality je žiadúca, jednotlivé časti diferenciálu $d\tilde{J}$, sú v každom okamihu k rovné 0. Z toho môžeme pre jednotlivé členy vytvoriť nasledujúce rovnice:

$$\begin{aligned} \nabla_{x_k} H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) - \vec{\lambda}_{k-1} &= 0, \\ \nabla_{u_k} H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) &= 0, \\ \nabla_{\lambda_k} H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) - \vec{x}_{k+1} &= 0, \\ (\nabla_{x_N} \Phi(\vec{x}_N, N) - \vec{\lambda}_{N-1})^T d\vec{x}_N &= 0, \\ (\nabla_{x_0} H_0(\vec{x}_0, \vec{u}_0, \vec{\lambda}_0))^T d\vec{x}_0 &= 0. \end{aligned} \quad (6.68)$$

Posledné dve rovnice ostali uvedené v diferenciálnom tvere nakoľko hodnota gradientu nemusí byť záväzne rovná nule. Preto možno o nich uvažovať nasledovne:

- Keďže počiatočné podmienky sú známe a fixné, aby bol výraz nulový, musí platiť $d\vec{x}_0 = 0$. To hovorí o skutočnosti, že počiatočná podmienka je konštantu a tým pádom možno túto rovnicu nahradiť definovaním počiatočných podmienok $\vec{x}_0 = \text{konst}$, kde konst je vektor pevne definovaných konštánt, v našom prípade stavy opisujúce magnet na počiatku levitácie.
- Koncové stavy \vec{x}_N , môžu byť buď známe, takže pre ne platí to isté, čo pre počiatočné stavy a teda \vec{x}_N je vektorom vopred zadefinovaných konštánt ($\vec{x}_N = \text{konst}$). Alebo neznáme, teda $d\vec{x}_N \neq 0$. Funkcia potom má určitú variáciu (jej hodnota je zmeniteľná vplyvom predošlých stavov) a musí platiť $\nabla_{x_N} \Phi(\vec{x}_N, N) - \vec{\lambda}_{N-1} = 0$.

Nakoľko je pri využívaní LQ riadenia reč o kvadratickom programovaní, je potrebné za doteraz využívané funkcie nájsť vhodnú substitúciu, zohľadňujúcu argumenty "kvadratickým" spôsobom. Nahraďme teda doteraz používané funkcie z Rov. (6.58) a (6.59) funkciami

$$\begin{aligned} \Phi(\vec{x}_N, N) &= \frac{1}{2}(\vec{x}_N^T S_N \vec{x}_N), \\ L_k(\vec{x}_k, \vec{u}_k) &= \frac{1}{2}(\vec{x}_k^T Q \vec{x}_k + \vec{u}_k^T R \vec{u}_k), \\ h_k(\vec{x}, \vec{u}) &= A\vec{x}_k + B\vec{u}_k. \end{aligned} \quad (6.69)$$

Tu by som rád priblížil význam matíc Q , R a S . Matica S , ako bola uvedená definícia funkcie $\Phi(\vec{x}_N, N)$, penalizuje výchylku koncového stavu od žiadanej referencie resp. od nuly. Matica Q má rovnaký význam, ale zohľadňuje túto odchýlku v čase pred dosiahnutím horizontu N . Posledná matica R má za úlohu penalizovať veľkosť vstupu a jeho podiel na celkovej hodnote účelovej funkcie. Pre matice Q a S platí, že čím väčšiu hodnotu majú, tým väčší dôraz sa kladie na minimalizáciu príslušných stavov. Pre maticu R platí, že čím väčšia hodnota, tým sa algoritmus snaží vytvárať menej agresívne vstupy. Voľba matíc Q a R je úlohou používateľa. Hodnota matice S je však objektom najbližších úvah, nakoľko práve tá a jej vývoj v čase sú zodpovedné za vytvorenie optimálneho riadiaceho zákona stabilizujúceho systém.

Po uvedenej substitúcií v Rov. (6.63) dostávame tvar Hemiltoniánu

$$H_k(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k) = \frac{1}{2}(\vec{x}_k^T Q \vec{x}_k + \vec{u}_k^T R \vec{u}_k) + \vec{\lambda}_k(A\vec{x}_k + B\vec{u}_k). \quad (6.70)$$

Následne sa nám zmení aj tvar Rov. (6.68) a po krátkej úprave dostávame set rovníc nazývaný aj "problém dvojbodových okrajových hodnôt" (z ang. *two-point boundary value problem*) [16, Kap.

5.2]:

$$\vec{\lambda}_{k-1} = Q\vec{x}_k + A^T \vec{\lambda}_k, \quad (6.71)$$

$$\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k, \quad (6.72)$$

$$R\vec{u}_k + B^T \vec{\lambda}_k = 0, \quad (6.73)$$

$$\vec{x}_0 = \vec{r}_0, \quad (6.74)$$

$$(S_N \vec{x}_N - \vec{\lambda}_{N-1})^T d\vec{x}_N = 0. \quad (6.75)$$

Prvé dve rovnice sú známe aj pod anglickými názvami (6.71) – *costate equation* a (6.72) – *state equation* a možno ich s využitím (6.73) zapísať do maticového tvaru:

$$\begin{bmatrix} \vec{x}_{k+1} \\ \vec{\lambda}_{k-1} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ Q & A^T \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{\lambda}_k \end{bmatrix}. \quad (6.76)$$

Teraz je voľba názvu problému očividnejšia. Okrajové podmienky sú zadané na dvoch rôznych miestach a to v počiatočnej a konečnej pozícii (Rov. (6.74) a Rov. (6.75)). Taktiež z Rov. (6.76) je zrejmé, že premenná $\vec{\lambda}_k$ sa vyvíja v čase späťne, zatiaľ čo \vec{x}_k sa mení s časom.

Teraz si predstavme, že chceme minimalizovať rozšírenú úcelovú funkciu $\tilde{J}(\vec{x}_k, \vec{u}_k, \vec{\lambda}_k)$ s tým, že koncový stav \vec{x}_N nie je vopred známy, takže pre Rov. (6.75) platí,

$$S_N \vec{x}_N = \vec{\lambda}_{N-1}. \quad (6.77)$$

Na základe toho možno vysloviť úvahu, že tento vzťah platí počas celého horizontu po okamih N a teda

$$\vec{\lambda}_{k-1} = S_k \vec{x}_k. \quad (6.78)$$

Hodnotu vstupu \vec{u}_k potom môžeme na základe Rov. (6.73) zapísať, ako funkciu budúceho stavu \vec{x}_{k+1} a matice S_{k+1} :

$$\vec{u}_k = -R^{-1}B^T S_{k+1} \vec{x}_{k+1}. \quad (6.79)$$

Substitúciou tohto výrazu do Rov. (6.72) a následne toho do Rov. (6.71) dostávame

$$S_k \vec{x}_k = Q\vec{x}_k + A^T S_{k+1} (I + BR^{-1}B^T S_{k+1})^{-1} A \vec{x}_k. \quad (6.80)$$

Zo vzťahu možno odstrániť \vec{x}_k (nakoľko je v každom člene) a dostávame “diferenčnú Riccatiho rovnicu v diskrétnom čase” [4]:

$$S_k = Q + A^T S_{k+1} (I + BR^{-1}B^T S_{k+1})^{-1} A. \quad (6.81)$$

Táto rovnica definuje dynamiku premennej S_k a to že sa vyvíja rekurentne, teda späťne v čase.

Teraz vstáva otázka, ako možno takýto výsledok využiť v tendenciách riadenia so spätnou väzbou. Ak rovnice prearanžujeme iným spôsobom a to, do Rov. (6.79) vložíme stavovú rovnicu (6.72) a osamostatníme \vec{u}_k , dostávame optimálny riadiaci LQ zákon [16, Kap. 5.3]:

$$\vec{u}_k^* = -\underbrace{(I + BR^{-1}B^T S_{k+1})^{-1} R^{-1} B^T S_{k+1} A}_{K_k} \vec{x}_k^*. \quad (6.82)$$

Problémom takého riadenia je ale to, že vstup \vec{u}_k^* je funkciou S_{k+1} , ktoré je definované riešením Rov. (6.81) a vyvíja sa v čase naspäť, čo nie je moc praktické. Naviac, pre výpočet by sme potrebovali mať aj fixný horizont, čo vo väčšine úloh riadenia nie je možné predpovedať.

Napriek tomu ale, ak vezmeme do úvahy, že vývoj premennej S_k je naozaj taký, že minimalizuje stavy \vec{x} pomocou vstupov \vec{u} , riešenie Riccatiho rovnice v Rov. (6.81) musí späť v čase konvergovať k určitej ustálenej hodnote. Ak posunieme koniec horizontu do nekonečna, teda $N \rightarrow \infty$, stabilným riešením Riccatiho rovnice bude konštantu S_∞ (ďalej len S). V rámci tejto úvahy možno Rov. (6.81) prepísať do tzv. "algebraickej Riccatiho rovnice" [4]¹²:

$$S = Q + A^T S (I + BR^{-1}B^T S)^{-1} A \quad (6.83)$$

a riadiaci zákon do podoby

$$\vec{u}_k = -\underbrace{(I + BR^{-1}B^T S)^{-1} R^{-1} B^T S A}_{K} \vec{x}_k, \quad (6.84)$$

kde S je riešením (6.83). Samotný výpočet Rov. (6.83) nie je triviálna záležitosť, využívajúca širšie poznatky lineárnej algebry a z tohto dôvodu, pri riešení uvedenej rovnice, som v rámci návrhu riadenia pre systém MagnetoShield použil funkciu `d1qr(A,B,Q,R)`, ktorá je súčasťou vývojového prostredia MATLAB.

Čo sa týka implementácie uvedenej teórie na praktický príklad riadenia procesu levitácie, nakoľko využívame totožný diskrétny model, ako ten opísaný v Rov. (6.42), našou jedinou úlohou ostáva vhodne zvoliť penalizačné matice Q a R pre LQ regulátor. Tu je základný rozdiel v tom, že už nemusíme navrhovať vlastné uloženie pôlov. Miesto toho uvažujme, ako veľmi je pre nás nutné nadobudnúť stabilitu jednotlivých stavov a koľko energie dovolíme systému na to použiť.

Algoritmus vždy navrhne stabilizujúce riešenie, ak je samotný systém stabilizovateľný (nemá žiadne neriaditeľné stavy¹³ a ak áno, samotné stavy sú asymptoticky stabilné [3]), jediné na čo si treba dať pozor je, aby sme príliš nezanedbali váhovanie vstupu R ($R \rightarrow 0$), nakoľko máme fyzikálne medze, v rámci ktorých sme schopní vstup vyprodukovať a prílišná benevolentnosť môže viesť k nezrealizovateľne veľkým vstupom, obzvlášť ak máme vysoké nároky na čas, za ktorý sa má dosiahnuť žiadaná referencia.

Samotný návrh matíc je väčšinou otázkou pokusov a opráv, aj keď je dobré vziať do úvahy niekoľko skutočností:

- Musí platiť $Q \succeq 0$, teda matica je pozitívna semi-definitná (ang. *positive semi-definite*). Je to z dôvodu, že ak by Q bola negatívna (negatívna semi-definitná), teda mala by aspoň jedno vlastné číslo > 0 , riešenie Rov. (6.76) pre $\vec{\lambda}_{k-1}$ by divergovalo¹⁴.
- Matica R musí byť pozitívna definitná teda $R \succ 0$, nakoľko v Rov. (6.76) robíme jej inverziu.
- Prvky na hlavnej diagonále penalizujú konkrétné stavy v \vec{x}_k , zatiaľ čo prvky mimo diagonály penalizujú určitú koreláciu medzi stavmi.

Vzhľadom k uvedenému som navrhol matice Q a R nasledovne:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 70 \end{bmatrix}, \quad R = [0.001].$$

¹²V literatúre častejšie uvádzaná v tvare po aplikácii "Identity Woodburyho matice" (ang. *Woodbury matrix identity*), teda $Q + A^T S A - S - A^T S B (B^T S B + R)^{-1} B^T S A = 0$

¹³Na overenie slúži matica riaditeľnosti Rov. (6.46)

¹⁴Toto platí per CT, v DT by riešenie divergovalo v prípade, že vlastné číslo $> |1|$

Táto voľba matíc odpovedá stabilizovaným pólov systému α a spätnovázobnému zosilneniu K :

$$\alpha = \begin{bmatrix} 0.3336 \\ 0.8715 \\ 0.9652 + 0.0339i \\ 0.9652 - 0.0339i \end{bmatrix}, \quad K = [-9691.17 \quad -206.31 \quad 84.02 \quad 208.59].$$

Rovnako, ako pri riadení uložením pólov, aj v tomto prípade je nutné navrhnúť pozorovateľ, ktorý bude odhadovať hodnotu stavov, ktoré v procese levitácie nedokážeme priamo merať. V tomto prípade použijeme Kalmanov filter opísaný v Kap. 6.3. Nakoľko som aj v prípade použitia Luenbergerovho pozorovateľa odhadoval stavy x_2 a x_3 (aj keď stav x_3 je možné merať), oba stavy budú odhadované aj v tomto prípade.

Kalmanov filter využíva rovnaké značenie pre kovariančné matice ako sme použili pri váhovacích maticiach LQ regulátora. Preto maticu kovariancie procesného šumu označíme Q_{kal} a meracieho šumu ako R_{kal} . Pri voľbe hodnôt Q_{kal} je dôležité vziať do úvahy skutočnosť, že model na základe ktorého robí Kalmanov filter odhad, je len identifikovaným priblížením k skutočnosti. Takže na hlavnej diagonále matice Q_{kal} môžu vyjsť pomerne veľké hodnoty. Pri matici R_{kal} je postup určovania variancie senzorov jednoduchší (kovariančná matica má na hlavnej diagonále hodnoty variancie pre danú veličinu).

Pre určenie, aká je variancia merania polohy pomocou Hallovho senzora, môžeme spraviť experiment. Na elektromagnet budeme posieláť konštantné napätie a počas tejto doby budeme merať polohu s určitou vzorkovacou periódou. Výsledok obsahuje polohu magnetu v čase a v ideálnom prípade by táto hodnota mala byť konštantná. Skutočnosť je ale iná a preto na výpočet variancie senzoru možno použiť funkciu v prostredí MATLAB `var()`, ktorá nám vráti žiadanú skalárnu hodnotu variancie testovaného senzoru. Rovnakým spôsobom sme schopní odhadnúť aj varianciu prúdového senzora¹⁵.

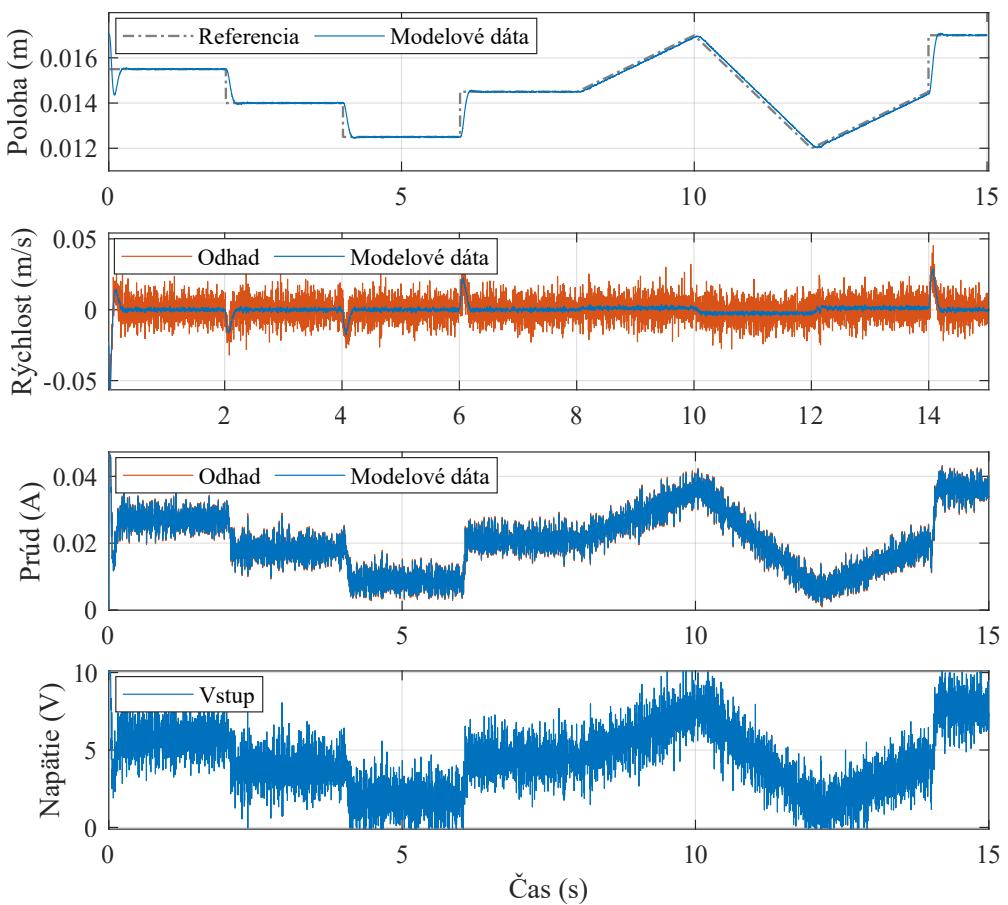
Výsledkom tohto postupu sú nasledovné hodnoty matíc Q_{kal} a R_{kal} , keď v prípade nastávajúceho riadenia budeme brať do úvahy použitie jedine Hallovho senzora:

$$Q_{\text{kal}} = \begin{bmatrix} 10^{-4} & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \quad R_{\text{kal}} = [1.58 * 10^{-10}]$$

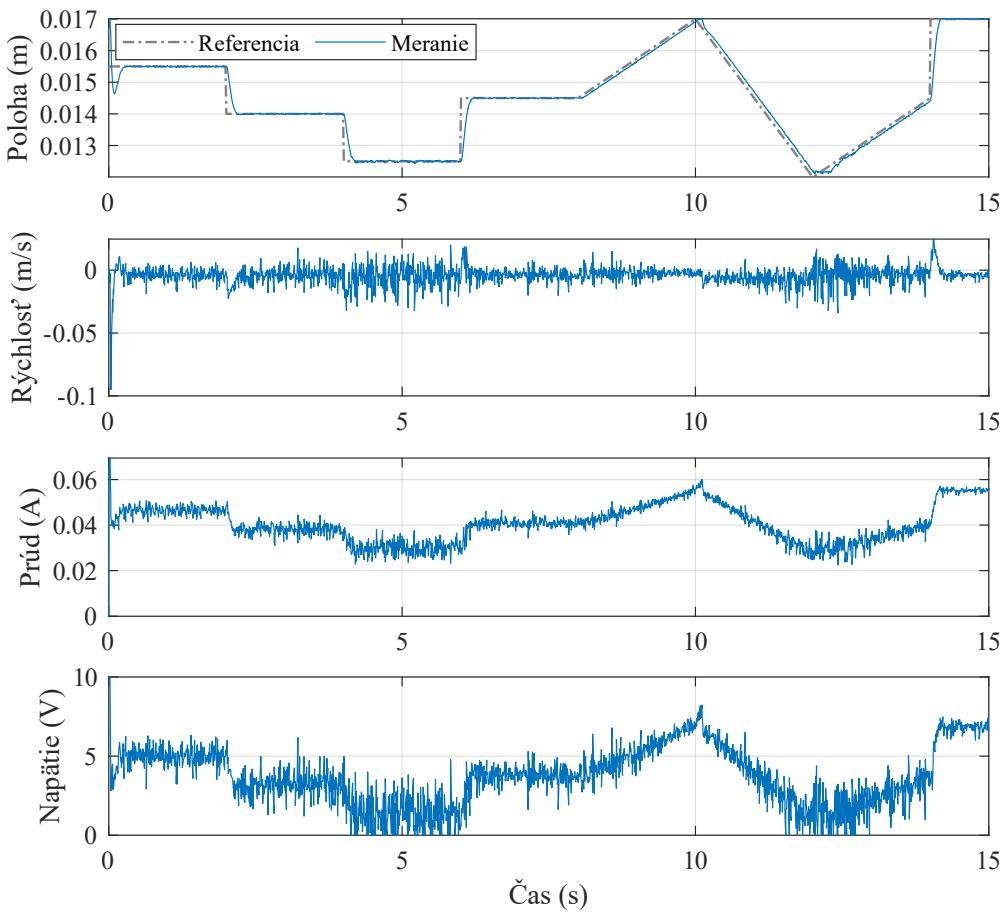
Navrhnuté ladenie systému som ako prvé odsimuloval v prostredí Simulink, mimo fyzického hardvéru a je zobrazené na Obr. 6.18. V simulácii je zahrnutý taktiež šum, rovnako ako v prípade simulácie riadenia uložením pólov s použitím Luenbergerovho pozorovateľa reprezentovanom na Obr. 6.14. Ak si porovnáme obe simulácie je na prvý pohľad zrejmé, že použitie Kalmanovho filtra vedie k menšej variácii odhadovaných veličín rýchlosťi a prúdu.

Výsledky experimentu po aplikácii navrhnutého riadenia na fyzický hardvér, pri perióde vzorkovania $T_s = 0.0015$ s, sú reprezentované na Obr. 6.19 a Obr. 6.20 zobrazuje riadiaci program vytvorený v prostredí Simulink. Nakoľko simulačné programy pre uloženie pólov a LQ riadenie sú si veľmi podobné, pre simuláciu tohto riadenia viď dokumenty v rámci otvoreného projektu MagnetoShield [19], časť "*simulink/examples/MagnetoShield*".

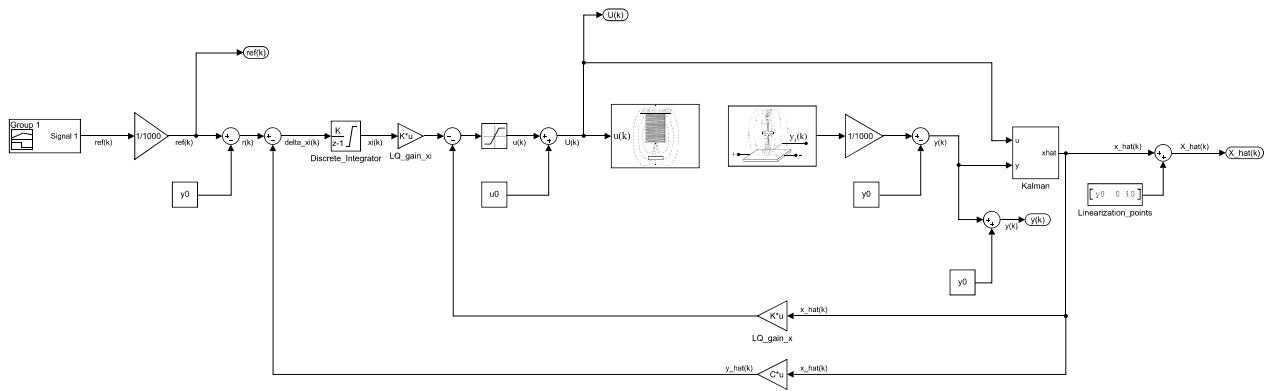
¹⁵Variancia pre Hallov senzor merajúci polohu vyšla $1.5815 * 10^{-10}$ m a pre prúdový senzor $6.6217 * 10^{-9}$ A.



Obr. 6.18: Simulácia riadenia LQ regulátorom s použitím Kalmanovho filtra uvažujúca prítomnosť šumu. „Modelové dátá“ reprezentujú simulovaný signál z modelu so zakomponovaným šumom, „Odhad“ je výstup z filtra.



Obr. 6.19: Priebeh riadenia LQ regulátorom s meraním jediného stavu a s použitím Kalmanovho filtra.



Obr. 6.20: Schéma riadenia LQ regulátorom.

6.7 Prediktívne riadenie

LQ regulácia má napriek svojmu širokému potenciálu uplatnenia taktiež jeden zásadný nedostatok. V Rov. (6.53) sme uviedli dva typy obmedzení a to obmedzenia rovnosti a nerovnosti. Následne sme však riešili optimalizačnú úlohu iba v tendenciách obmedzení rovnosti, ktoré reprezentujú rovnice stavového modelu. Po zostavení účelovej funkcie a zarátaní týchto obmedzení pomocou Lagrangerových multiplikátorov, sme na základe nutnej podmienky optimality prvého rádu vytvorili problém dvojbodových okrajových hodnôt. Ten nám udával vývoj váhovania koncového stavu S_k v čase, ako riešenie diferenčnej Riccatiho rovnice. Po úvahе, že riešenie S_k konverguje k stabilnej hodnote, sme vypočítali S_∞ a jeho pomocou vyjadrili pevný riadiaci zákon ako funkciu S_∞ a aktuálneho stavu \vec{x}_k . Avšak, ak existujú určité obmedzenia na vstupnú veličinu (napr. pre MagnetoShield maximálne vstupné napätie), v prípade LQ riadenia nie sú súčasťou optimalizačnej úlohy a vstup je nutné saturovať. To ale nie je riešením problému, lebo o saturácií regulátor nevie a v prípade požiadavky rýchleho riadenia to vedie k nestabilite systému. Snaha o optimálne riadenie spolu s aplikáciou obmedzení nerovnosti vedie k novému typu riadenia nazývanému prediktívne riadenie alebo tiež MPC (z ang. *model predictive control*).

Riešenie optimalizačnej úlohy s ohraničeniami v tvare nerovnosti udáva nevyhnutná podmienka optimality prvého rádu, nazávaná aj "podmienka Karush–Kuhn–Tucker", tzv. KKT podmienka [10, 22] (jedná sa o to tú istú podmienku ako bola v Kap. 6.6, len v univerzálnej forme). Úloha tvaru

$$\begin{aligned} \min_{\vec{x}} J(\vec{x}), \\ h_i(\vec{x}) = 0, \\ g_j(\vec{x}) \leq 0, \end{aligned} \tag{6.85}$$

má optimálne riešenie \vec{x}^* v prípade, že platí

$$\begin{aligned} \nabla J(\vec{x}^*) + \sum_{i=1}^m \lambda_i \nabla h_i(\vec{x}^*) + \sum_{j=1}^n \rho_j \nabla g_j(\vec{x}^*) = 0, \\ \rho_j g_j(\vec{x}^*) = 0, \\ \rho_j \geq 0. \end{aligned} \tag{6.86}$$

Okrem už spomenutého Lagrangeového multiplikátora λ_i (teória a jeho význam tu platí úplne totožne ako v predchádzajúcej kapitole), tu máme aj konštantu ρ_j .

Z Rov. (6.85) vieme, že aby sme splnili podmienku nerovnosti, funkcia $g_j(\vec{x})$ musí mať nulovú alebo zápornú hodnotu. Od toho sa odvíja aj význam ρ_j :

- V prípade, že je hodnota nulová, dívame sa na túto podmienku, ako na podmienku rovnosti $g_j(\vec{x}) = 0$ a teda ρ_j je klané nenulové číslo a funguje ako Lagrangeov multiplikátor λ .
- Ak však riešime bod, ktorý leží vnútri oblasti priateľných riešení a platí preňho $g_j(\vec{x}) < 0$, podmienka nerovnosti je splnená, tým pádom $\rho_j = 0$ a potrebnou podmienkou optimality, za predpokladu, že existuje obmedzenie v tvare rovnosti $h_i(\vec{x}) = 0$, je len výraz

$$\nabla J(\vec{x}^*) + \sum_{i=1}^m \lambda_i \nabla h_i(\vec{x}^*) = 0. \tag{6.87}$$

Teraz je zrejmý aj význam rovnosti $\rho_j g_j(\vec{x}^*) = 0$ v Rov. (6.86). Je tam z dôvodu "rozhodovania", či má byť ρ_j nulové alebo nie.

Opísaný princíp možno intuitívne aplikovať pomocou nasledovného algoritmu riešenia takýchto úloh. Predstavme si, že máme minimalizačnú úlohu tvaru:

$$\begin{aligned} \min_{\vec{x}} J(\vec{x}), \\ g_1(\vec{x}) \leq 0, \\ g_2(\vec{x}) \leq 0. \end{aligned} \tag{6.88}$$

Pri tejto metóde postupujeme nasledovne:

1. Vypočítame minimum pre funkciu $J(\vec{x})$ bez obmedzení (riešime systém rovníc $\nabla J(\vec{x}^*) = 0$).
2. Vypočítame minimum pre úlohu s obmedzením $g_1(\vec{x}) = 0$ (riešime systém rovníc $\nabla J(\vec{x}^*) + \rho_1 \nabla g_1(\vec{x}^*) = 0$, pre $\rho_1 > 0$).
3. Vypočítame minimum pre úlohu s obmedzením $g_2(\vec{x}) = 0$.
4. Vypočítame minimum pre úlohu s obmedzeniami $g_1(\vec{x}) = 0$ a $g_2(\vec{x}) = 0$ (ak by bod ležal na priesecíku oboch obmedzení).
5. Vyberieme z riešení také minimum, ktoré splňa obe nerovnosti.

Je zrejmé, že s narastajúcim počtom obmedzení sa bude počet čiastkových úloh zväčšovať spôsobom 2^n , kde n je počet obmedzení, čo je veľmi neefektívne. Preto na riešenie takýchto úloh sú vyvynuté iné numerické metódy. Tento postup je ale veľmi intuitívny a poskytuje skvelý náhľad na to, ako sa s problémom vysporiadať.

Okrem toho nám Rov. (6.86) odhaluje jednu zásadnú vlastnosť pri riešení optimalizačnej úlohy s obmedzeniami nerovnosti. Tieto 3 výrazy, charakterizujúce optimálne riešenie nevieme jednotne sformulovať do pevného spätnoväzobného riadiaceho zákona opísaného jedinou rovnicou (teda zákon s formou $u_k = -Kx_k$).

Principiálne, spomenutá teória by išla priamo aplikovať na kvadratickú úcelovú funkciu

$$J(\vec{x}, \vec{u}) = \frac{1}{2} \vec{x}_N^T S \vec{x}_N + \frac{1}{2} \sum_{k=0}^{N-1} (\vec{x}_k^T Q \vec{x}_k + \vec{u}_k^T R \vec{u}_k), \tag{6.89}$$

s akou sme pracovali v minulej kapitole. Avšak skutočnosť, že S je už známe (vieme vypočítať S_∞) a obmedzenie rovnosti (stavový systém) nám definuje budúci stav $\vec{x}_{k+1} = A\vec{x}_k + B\vec{u}_k$, možno túto úcelovú funkciu preformulovať do tvaru:

$$\begin{aligned} J(\vec{x}, \vec{u}) = & \frac{1}{2} \left(\underbrace{\begin{bmatrix} \vec{x}_1^T & \dots & \vec{x}_N^T \end{bmatrix}}_{\vec{x}^T} \underbrace{\begin{bmatrix} Q & 0 & 0 & \dots \\ 0 & Q & & \\ 0 & & \ddots & 0 \\ \vdots & 0 & S \end{bmatrix}}_{Q} \begin{bmatrix} \vec{x}_1 \\ \vdots \\ \vec{x}_N \end{bmatrix} \right. \\ & \left. + \underbrace{\begin{bmatrix} \vec{u}_0^T & \dots & \vec{u}_{N-1}^T \end{bmatrix}}_{\vec{u}^T} \underbrace{\begin{bmatrix} R & 0 & \dots \\ 0 & R & \\ \vdots & & \ddots \end{bmatrix}}_R \begin{bmatrix} \vec{u}_0 \\ \vdots \\ \vec{u}_{N-1} \end{bmatrix} + \vec{x}_0^T Q \vec{x}_0 \right). \end{aligned} \tag{6.90}$$

Úcelová funkcia je stále funkciou premenných \vec{x} a \vec{u} , ktoré po novom obsahujú všetky stavy a všetky vstupy počas horizontu, ale vďaka stavovému systému sme schopní z nej spraviť funkciu

vstupu \vec{u} a počiatočného stavu \vec{x}_0 . Ako prvé vyjadrimo podmienku rovnosti stavového systému ako funkciu \vec{x}_0 :

$$\begin{aligned}\vec{x}_1 &= A\vec{x}_0 + B\vec{u}, \\ \vec{x}_2 &= A\vec{x}_1 + B\vec{u} = A(A\vec{x}_0 + B\vec{u}) + B\vec{u}, \\ &\dots, \\ \vec{x}_k &= A^k\vec{x}_0 + [A^{k-1}B \quad A^{k-2} \quad \dots \quad AB \quad B]\vec{u}.\end{aligned}\tag{6.91}$$

V maticovom tvare to môžeme zapísť ako

$$\vec{x} = \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & & 0 \\ \dots & & \ddots & \dots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\bar{N}} \vec{u} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{M}} \vec{x}_0.\tag{6.92}$$

Teraz vložme Rov. (6.92) do účelovej funkcie (6.90). Výsledkom je funkcia, ktorá je závislá už len od počiatočného stavu \vec{x}_0 a vstupu:

$$\bar{J}(\vec{u}, \vec{x}_0) = \frac{1}{2}((\bar{N}\vec{u} + \bar{M}\vec{x}_0)^T \bar{Q}(\bar{N}\vec{u} + \bar{M}\vec{x}_0) + \vec{u}^T \bar{R}\vec{u} + \vec{x}_0^T \bar{Q}\vec{x}_0).\tag{6.93}$$

Z tohto, po niekoľkých úpravách a odstránení členov, ktoré sú závislé len na \vec{x}_0 (napr. člen $\vec{x}_0^T \bar{Q}\vec{x}_0$ je konštantu a nevieme ho optimalizačným procesom ovplyvniť, tým pádom je v účelovej funkcií nepotrebný), dostávame novú formuláciu účelovej funkcie, v ktorej minimalizujeme jedinú premennú a tou je \vec{u} [22, Kap. 4.3]:

$$\bar{J}(\vec{u}) = \frac{1}{2}\vec{u}^T H\vec{u} + \vec{x}_0^T G^T \vec{u},\tag{6.94}$$

resp.

$$\bar{J}(\vec{u}) = \vec{u}^T H\vec{u} + 2\vec{x}_0^T G^T \vec{u},\tag{6.95}$$

kde matica H sa nazýva "Hessián" a výraz $\vec{x}_0^T G^T$ sa volá "gradient" a platí pre ne:

$$\begin{aligned}H &= \bar{N}^T \bar{Q} \bar{N} + \bar{R}, \\ G^T &= \bar{M}^T \bar{Q} \bar{N}.\end{aligned}\tag{6.96}$$

Výhodnosť tohto tvaru spočíva v tom, že obmedzenie rovnosti máme zahrnuté v účelovej funkcií, matice H a G možno vypočítať "offline" pred chodom algoritmu a z optimalizačnej úlohy, akú som sformuloval v Rov. (6.85) sa stáva úloha ohraničená jedine nerovnosťami.

Ďalej je nutné uviesť formuláciu obmedzení nerovnosti. Najčastejším typom takéhoto obmedzenia sú maximálne hodnoty vstupnej veličiny \vec{u} . Vzhľadom na požiadavku z Rov. (6.85) $g_j(\vec{x}) \leq 0$, možno písat

$$\begin{aligned}\vec{u} \leq \vec{u}_{max} &\rightarrow \begin{bmatrix} \vec{u}_0 \\ \vdots \\ \vec{u}_{N-1} \end{bmatrix} \leq \begin{bmatrix} \vec{u}_{max} \\ \vec{u}_{max} \\ \vdots \end{bmatrix}, \\ -\vec{u} \leq -\vec{u}_{min} &\rightarrow -\begin{bmatrix} \vec{u}_0 \\ \vdots \\ \vec{u}_{N-1} \end{bmatrix} \leq -\begin{bmatrix} \vec{u}_{min} \\ \vec{u}_{min} \\ \vdots \end{bmatrix},\end{aligned}\tag{6.97}$$

kde treba vziať do úvahy, že \vec{x}_k a $\vec{u}_{max/min}$ môžu byť vektory (napr. pre MIMO systém). Obdobne možno sformulovať aj obmedzenia stavov. Tu sa dá využiť vyjadrenie k-teho stavu pomocou počiatočného stavu \vec{x}_0 z Rov. (6.92):

$$\begin{aligned}\vec{x}_k \leq \vec{x}_{max} &\rightarrow \bar{M}\vec{x}_0 + \bar{N}\vec{u} \leq \vec{x}_{max}, \\ -\vec{x}_k \leq -\vec{x}_{min} &\rightarrow -\bar{M}\vec{x}_0 - \bar{N}\vec{u} \leq -\vec{x}_{min}.\end{aligned}\quad (6.98)$$

Tu je vidno, že obmedzenie stavov je opäť len obmedzením vstupov, vyjadrené vhodným spôsobom (nakol'ko $N\vec{u}$ možno osamostatniť). Takže optimalizačná úloha, ktorá je riešená v rámci MPC algoritmu môže mať formuláciu:

$$\begin{aligned}\min_{\vec{u}} \quad & \vec{u}^T H \vec{u} + 2\vec{x}_0^T G^T \vec{u}, \\ \vec{u} \leq & \vec{u}_{max}, \\ -\vec{u} \leq & -\vec{u}_{min}, \\ \bar{N}\vec{u} \leq & \vec{x}_{max} - \bar{M}\vec{x}_0, \\ -\bar{N}\vec{u} \leq & -\vec{x}_{min} + \bar{M}\vec{x}_0.\end{aligned}\quad (6.99)$$

Posledná nutná teoretická poznámka k prediktívnemu riadeniu je skutočnosť, že uvedená úloha je riešená v každej iterácii riadenia. Stavy \vec{x}_0 predstavujú aktuálne namerané veličiny a vektor \vec{u} obsahuje všetky potrebné vstupy do systému na to, aby sa dosiahol optimálny priebeh riadenia, až po koniec horizontu N . Avšak, nakol'ko model systému, na základe ktorého sa vstupy počítajú nie je dokonalý a nevieme v ňom obsiahnuť úplnú zložitosť systému aj s náhodnými faktormi, ktoré naň pôsobia, vždy sa používa jedine prvý vstup \vec{u}_0 zo sekvencie vstupov \vec{u} .

Pri aplikácii MPC na fyzický systém musíme opäť navrhnúť matice Q a R , ktoré principiálne plnia totožnú úlohu ako pri LQ riadení a to, že penalizujú nenulovú hodnotu stavov (matica Q) a vstupov (matica R). Okrem toho však je nutné zvoliť aj tretí parameter a tým je dĺžka predikčného horizontu N , teda koľko krokov dopredu berie algoritmus ohľad. Je zrejmé, že čím dlhší horizont, tým lepšie riadenie vieme dosiahnuť (vieme, aký dopad bude mať aktuálna veľkosť vstupu na stavy v budúcnosti). Avšak zariadenie MagnetoShield je primárne určené pre prototypizáčné dosky Arduino, ktoré majú obmedzený výpočtový výkon a teda je nutné navrhnúť takú dĺžku horizontu, že budeme schopní dodržať žiadnanú vzorkovaciu períodu¹⁶.

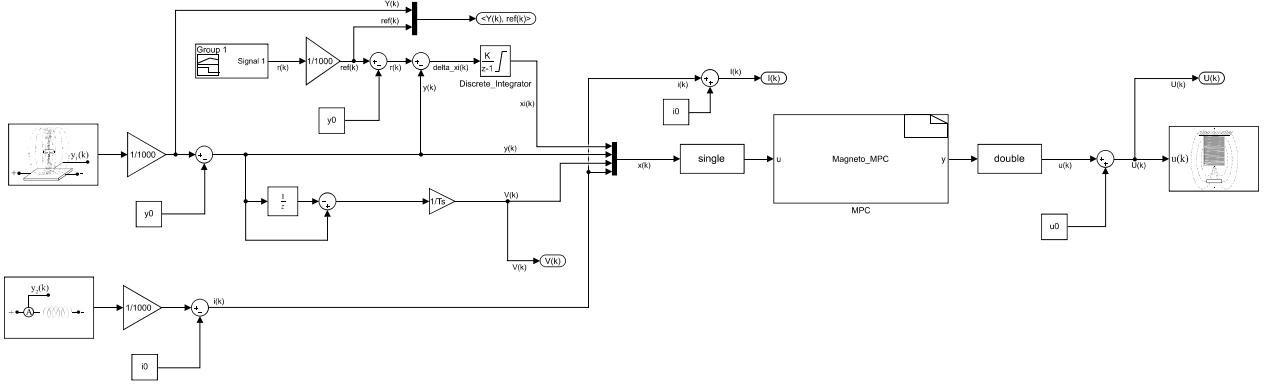
Pri zohľadnení uvedeného, som pre ukážku zvolil nasledovné ladenie systému:

$$Q = \begin{bmatrix} 240 & 0 & 0 & 0 \\ 0 & 260 & 0 & 0 \\ 0 & 0 & 80 & 0 \\ 0 & 0 & 0 & 800 \end{bmatrix}, \quad R = [0.01], \quad N = 20.$$

Opäť máme jedinú vstupnú veličinu a tou je napätie. Taktiež štyri stavy rozšíreného stavového modelu, opísaného diskrétnym modelom v Rov. (6.42).

Nakoľko existuje maximálne vstupné napätie, ktoré sme schopní na elektromagnete nastaviť (10 V) je potrebné túto skutočnosť vyjadriť v podobe obmedzenia nerovnosti. Tu je ešte potrebné pripomenúť, že identifikovaný model, ktorý pri riadení používame, neuvažuje priamo napätie, ale

¹⁶Povedzme, že máme obmedzenie na vstup v podobe $\vec{u}_{min} \leq \vec{u} \leq \vec{u}_{max}$. Potom pre každú +1 veľkosť horizontu získame +2 obmedzenia. Ak vezmeme do úvahy metódu opísanú na začiatku podkapitoly, pre systém s horizontom $N = 10$ máme 20 obmedzení, čo by znamenalo minimálne $2^{20} = 1048576$ výpočtových úloh potrebných pre nájdenie optimálneho vstupu touto metódou.



Obr. 6.21: Schéma riadenia polohy levitácie na základe prediktívneho riadenia.

jeho odchýlku od linearizačnej hodnoty u_0 definovanej v Rov. (5.34) (pozor, tu je u_0 hodnota fyzikálnej veličiny, nie nultý člen postupnosti vstupov \vec{u}_k). Takže pre vstupné napätie musí platiť

$$0 - u_0 \leq \vec{u}_k \leq 10 - u_0. \quad (6.100)$$

V prípade linearizačnej hodnoty, ktorá bola použitá pri identifikácii lineárneho modelu, $u_0 = 5.0056$ V, dostávame podmienku nerovnosti v tvare:

$$-5.0056 \text{ V} \leq \vec{u}_k \leq 4.9944 \text{ V}. \quad (6.101)$$

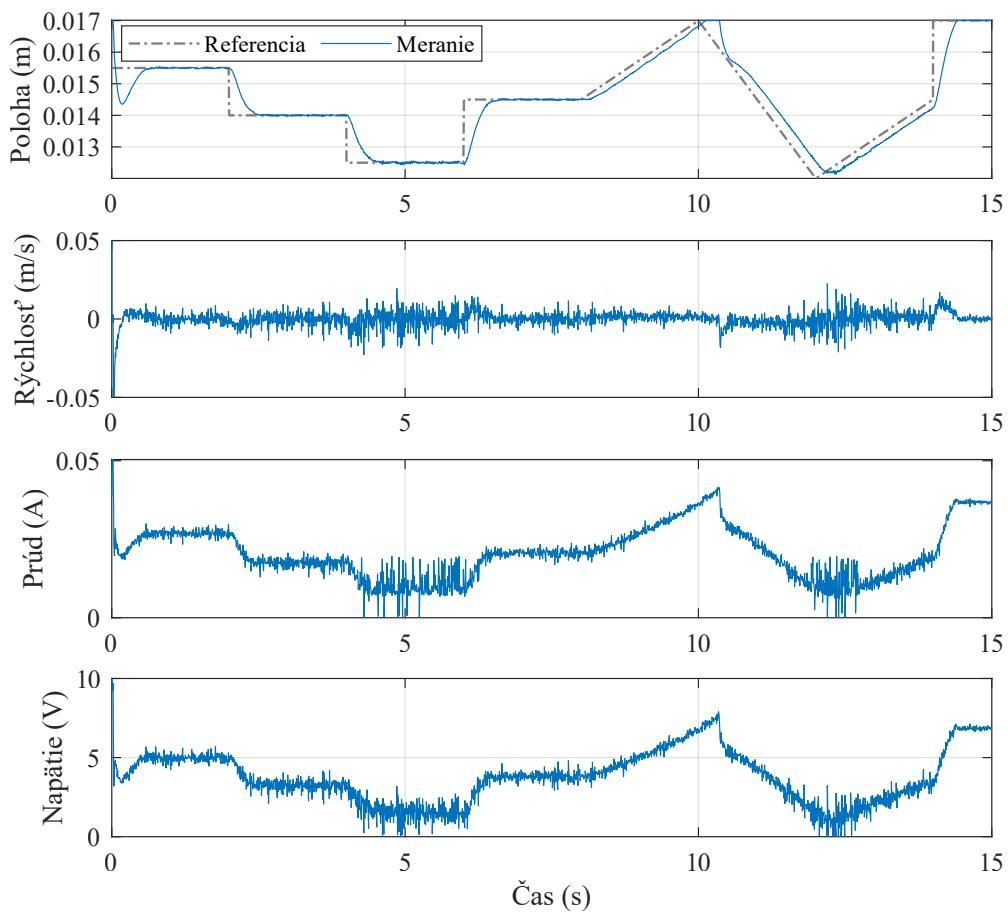
V rámci návrhu riadiaceho programu využívajúceho MPC sú meranými stavmi poloha a aj elektrický prúd. Nakoľko uvedený algoritmus je výpočtovo náročný, pre odhad rýchlosť som použil spätnú Eulerovu metódu opísanú Rov. (6.19), ktorej výpočtové požiadavky sú minimálne. Na druhej strane, pri odhade stavu sa šum merania polohy prenáša razantným spôsobom aj do odhadu rýchlosť. Výsledný riadiaci program zostrojený v prostredí Simulink je na Obr. 6.21.

Ako som uviedol, prediktívne riadenie je typ algoritmu, ktorý nie je možné sformulovať v podobe pevného riadiaceho zákona. Z tohto dôvodu je algoritmus do prostredia Simulink implementovaný, ako externý kód napísaný v jazyku C. Tento kód je výsledkom použitia prevzatého softvéru na modelovanie a riešenie kvadratickej optimalizačnej úlohy $\mu AO-MPC$ [31]. Na automatické generovanie kódu v jazyku C je využívaný inicializačný skript v jazyku MATLAB „MagnetoShield_Export_muAOOMP“C“, ktorý je možné nájsť na stránke projektu [19] v časti „matlab/examples/MagnetoShield“ alebo v časti „simulink/examples/MagnetoShield/MagnetoShield_AlternativeModel_Examples“. Ten definuje optimalizačnú úlohu z hľadiska opisu stavového systému, obmedzení nerovnosti a nastavení jednotlivých váhovacích matíc Q a R , dĺžky horizontu N a períody vzorkovania T_s . Jeho spustením sa vygenerujú dva nové skripty v jazyku Python, ktoré slúžia ako vstup pre program $\mu AO-MPC$. Ich prehratím a spracovaním sa zostaví algoritmus na riešenie konkrétnej úlohy v jazyku C. Vytvorené C súbory je možné nájsť v priečinku s preddefinovaným názvom *cmpc*. Ich implementácia do prostredia Simulink je zabezpečená pomocou bloku *S-Function Builder* (na Obr. 6.21 blok *Magneto_MPC*). Pre prípadné zmeny v rámci bloku odporúčam prečítať jeho dokumentáciu.

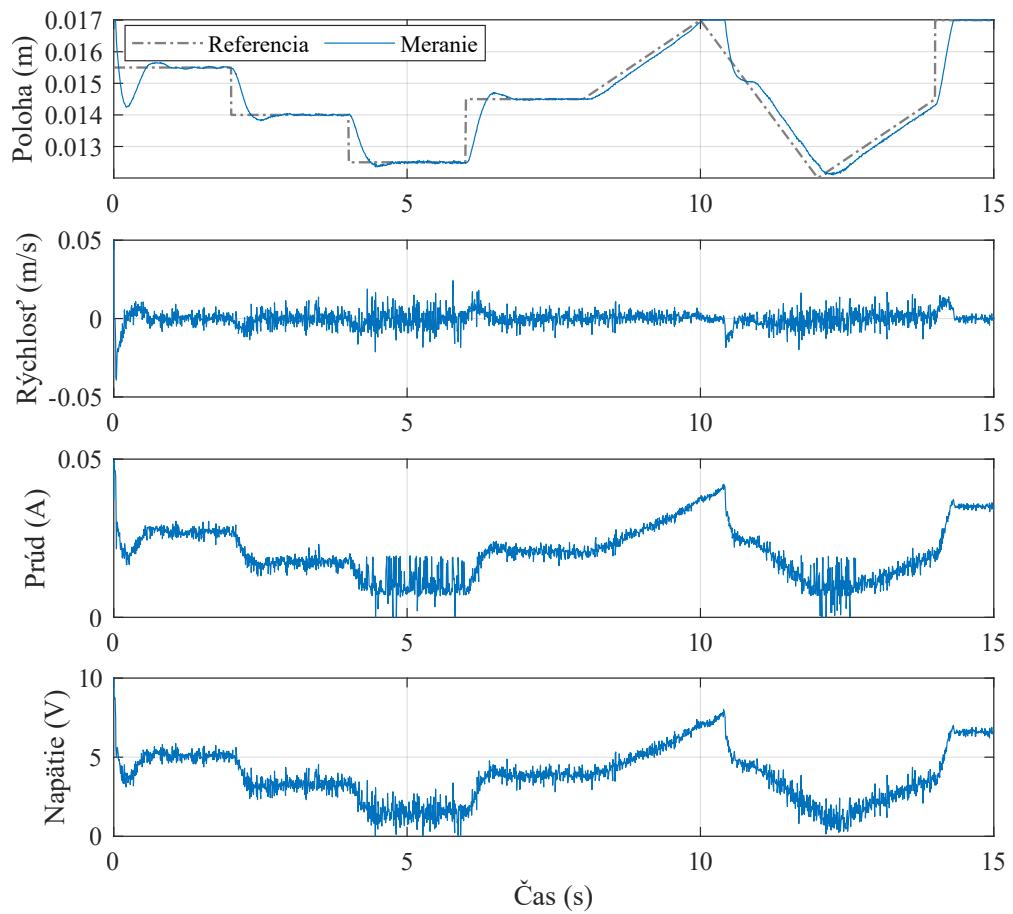
V prípade zmeny systému, alebo jedného z jeho parametrov je potrebné využívať skript definujúci systém v jazyku MATLAB, alebo vykonať úpravy priamo v Python skriptoch. Následne je, pre úspešnú implementáciu do fyzického systému, nutné dodržať uvedený postup:

1. Vykonáť požadované zmeny v MATLAB skripte “*MagnetoShield_Export_muAOMPC*” a spustiť skript.
2. Z cesty v MATLABe odstrániť priečinok *cmpc*, ak sa tam predtým nachádzal (v prostredí MATLAB “HOME/Set Path”).
3. Spustiť vygenerovaný skript v jazyku Python “*MagnetoShield_muAOMPC_Main.py*”.
4. Zahrnúť novo vygenerovaný priečinok *cmpc* do cesty MATLABu.
5. Do riadiaceho programu vytvorenom v Simulinku, zahrnúť vytvorené súbory z *cmpc* do bloku *S-Function Builder* a stlačiť *Build* (vytvorí to potrebné prepojenie medzi algoritmom v C a Simulinkom).
6. Následne je možné program nahrať ho do Arduina.

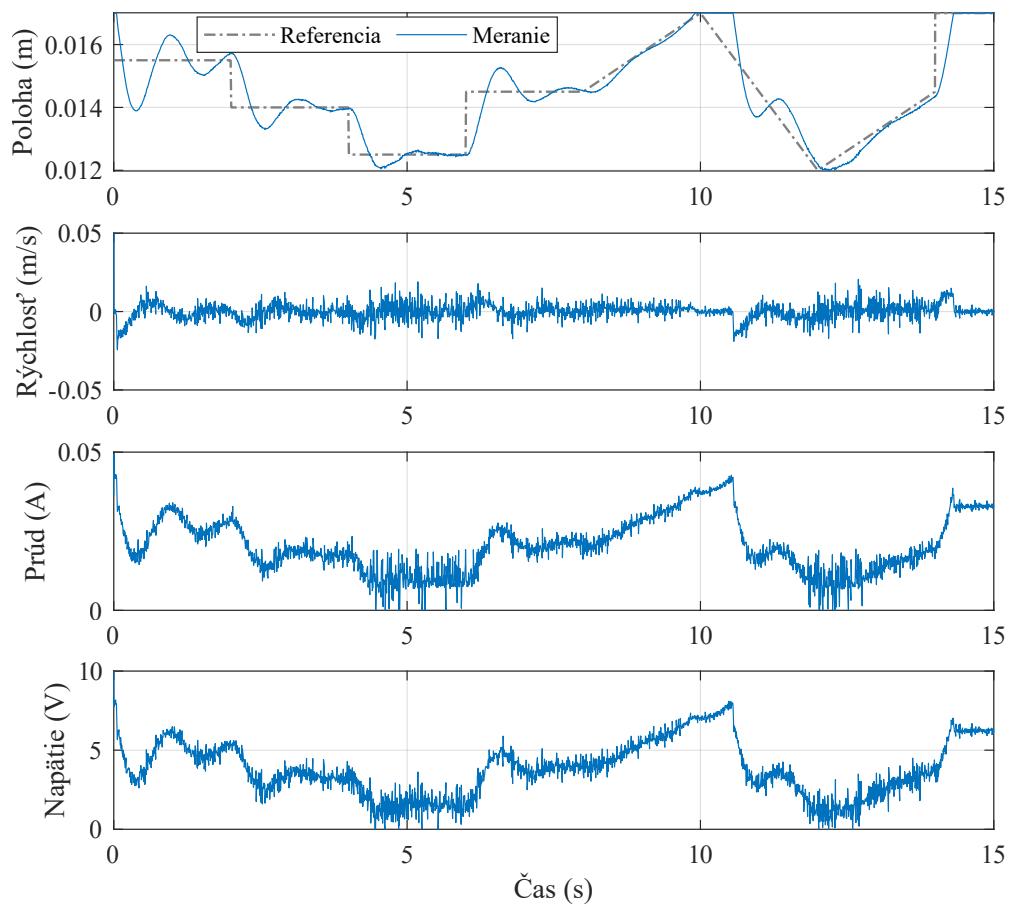
Výsledkom opísaných krokov je riadenie s použitím MPC, ktorého priebeh je možné vidieť na Obr. 6.22. Okrem toho sú na Obr. 6.23 a 6.24 zobrazené priebehy levitácie pre ladenie systému s rôznou dĺžkou horizontu.



Obr. 6.22: Priebeh stavových veličín a vstupu pri MPC riadení s predikčným horizontom $N = 20$.



Obr. 6.23: Priebeh stavových veličín a vstupu pri MPC riadení s predikčným horizontom $N = 15$.



Obr. 6.24: Priebeh stavových veličín a vstupu pri MPC riadení s predikčným horizontom $N = 10$.

Nevýhodou MPC je jeho výpočtová náročnosť, ktorá výrazne rastie s pribúdajúcimi obmedzeniami a zväčšujúcim sa horizontom. Existuje niekoľko spôsobov ako algoritmus MPC zjednodušíť a tým dosiahnuť vyššie vzorkovacie frekvencie. Jedným zo spôsobov je tzv. explicitné prediktívne riadenie alebo tiež eMPC, ktoré spočíva v “*off-line*” predpočítaní rôznych lineárnych riadiacich zákonov v závislosti od momentálneho stavu \vec{x}_0 . Následne nie je úlohou riadenia vypočítať optimalizačnú úlohu, ale nájsť vhodný riadiaci zákon (región) na základe stavu [14]. Tento typ riešenia v prípade menších systémov výrazne znižuje výpočtový čas, ale je náročnejší na pamäť a uschovanie vopred vypočítaných riešení.

V rámci mojej diplomovej práce som pre “*off-line*” riešenie využíval knižnicu *Multi-Parametric Toolbox 3* [9] určenú pre MATLAB a vygenerované výsledky som aplikoval v podobe riadiaceho programu v prostredí Simulink.

Pre využívanie eMPC v rámci iniciatívy AutomationShield je napísaný MATLAB skript s názvom “*MagnetoShield_Export_EMPC*”. V rámci tohto skriptu sa nastavuje vzorkovanie, použitý model, ladenie regulátora (matice Q a R), dĺžka horizontu a exportovanie riadenia. V rámci exportovania riadenia je možné zvoliť, pre aký výpočtový softvér resp. jazyk chceme, aby bol algoritmus zostavený (nežiadúce nastavenia je dobré zakomentovať).

V rámci riadiacej schémy zobrazenej na Obr. 6.26, je zostavený algoritmus pridaný do riadenia pomocou bolku *S-Function*, do ktorého je nutné vložiť názov tzv. MEX funkcie, ktorá je po spustení skriptu vygenerovaná spolu s ďalšími dodatočnými súbormi.

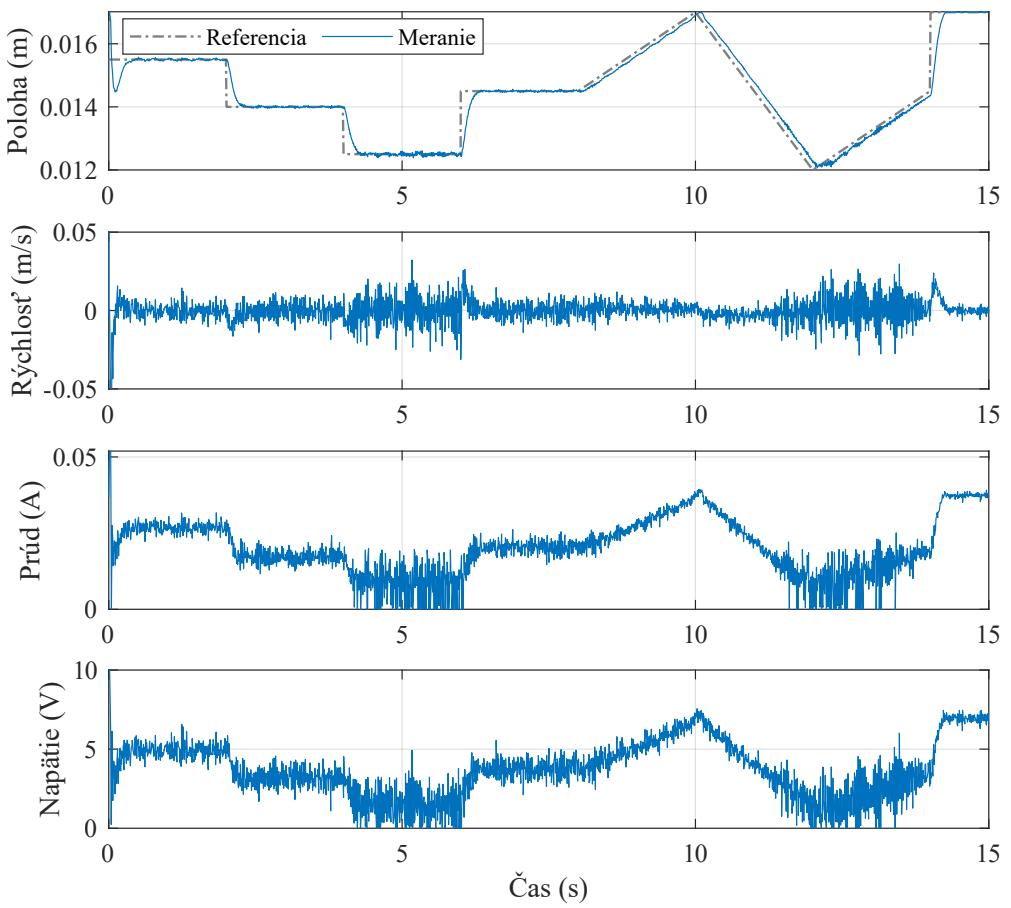
Výstup z bloku je vždy celý predikčný horizont, takže v prípade horizontu $N = 5$ to bude vektor o dĺžke 5 elementov. Z toho dôvodu sa hneď za výstupom nachádza vektor násobiaci tento výstup a vyberajúci jedine prvý člen vektora (vstup \vec{u}_0).

Pre implementáciu explicitného prediktívneho riadenia na fyzický systém MagnetoShield som zvolil nasledovné hodnoty matíc a horizontu:

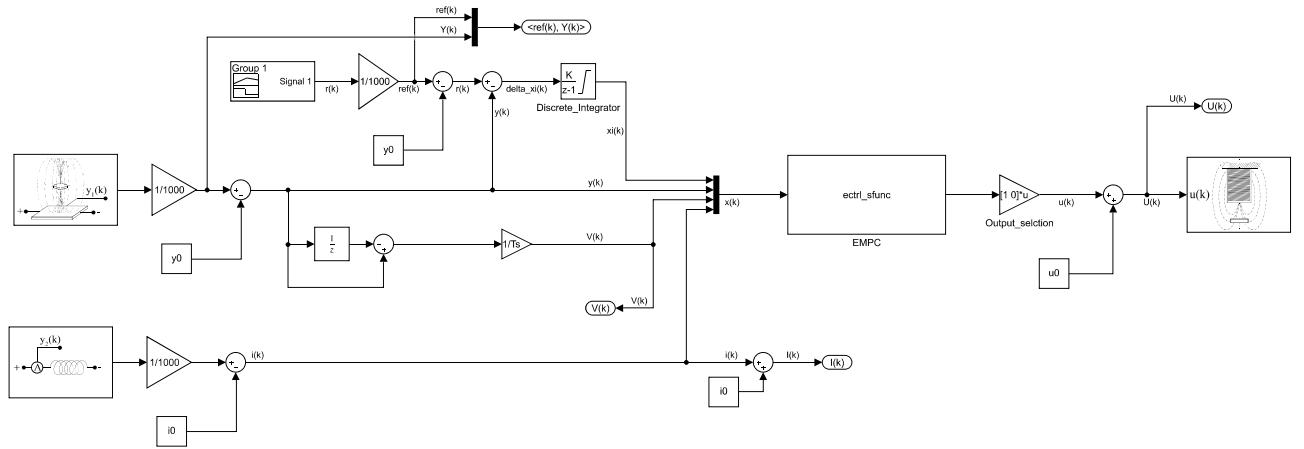
$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 40 & 0 \\ 0 & 0 & 0 & 70 \end{bmatrix}, \quad R = [0.01], \quad N = 2.$$

Nakoľko pamäťové a výpočtové kapacity prototypizačnej dosky Arduino nie sú vysoké, zvolil som krátky horizont v dôsledku čoho som sa rozhodol pre totožné ladenie, aké bolo použité pre LQ regulátor. Výsledný priebeh levitácie s použitím eMPC je zobrazený na Obr. 6.25.

Pri porovnaní MPC (Obr. 6.22) s eMPC (Obr. 6.25) je vidieť rozdiel pri kontinuálnej zmene referencie, že eMPC v čase $t = 10$ s lepšie sleduje trajektóriu (rovako aj LQ riadenie). Tento fenomén je ale daný rozdielnosťou ladenia oboch regulátorov. V prípade sledovania ustálenej referencie (časový interval 0 až 7 s) sú priebehy porovnateľné s tým, že MPC vykazuje menšiu variáciu vstupnej veličiny – napätia.



Obr. 6.25: Priebeh riadenia pomocou eMPC.



Obr. 6.26: Schéma riadenia pomocou eMPC.

7 Dodatočné príklady

V rámci predchádzajúcej kapitoly som uviedol teoretické základy pre správne navrhnutie systému s použitím riadenia založeného na jeho stavovom modeli a taktiež som tieto základy aplikoval pri návrhu fungujúceho riadenia. Avšak, napriek tomu sa vynára niekoľko otázok, ktoré úzko súvisia s riadením, ale neboli zodpovedané. Napríklad, aký je reálny rozdiel medzi dvomi spomenutými modelmi v kvalite opisu systému, nakoľko dobre vystihuje model podstatu systému v nelineárnom tvaru a taktiež, či je možné naraz riadiť oba merateľné stavy, polohu a prúd pri sledovaní ich nezávislých referencií? Na tieto otázky sa posnažím v tejto kapitole podať stručnú odpoveď.

7.1 Modely systému MagnetoShield

V Kap. 5 som sa venoval vlastnému návrhu modelu systému MagnetoShield. Už pri návrhu a predbežných výpočtoch sa dosiahla veľmi vysoká percentuálna zhoda modelu a nameraných dát, zozbieraných pomocou LQ riadenia (Obr. 5.9). Následne bol model identifikovaný a jeho presnosť sa ešte vylepšila (Obr. 5.10). Riadenie využité pri zbere dát ale už využívalo model a to ten, opísaný v článku „*MagnetoShield: Prototype of a Low-Cost Magnetic Levitation Device for Control Education*“ [20]. To by mohlo vyvolávať dohad, či je naozaj nový model taký presný, alebo je percentuálna zhoda len zdánlivá a je taká vysoká len pri danej vzorke dát (samozrejme treba brať ohľad na to, že pri každej vzorke dát, bude zhoda modelu s reálnym meraním iná, nakoľko každý experiment má jedinečný priebeh)?

V snahe odpovedať na túto otázkou som realizoval nezávislé porovnanie oboch modelov na dátach, ktoré boli zozbierané za pomoci PID riadenia na tom istom zariadení pri totožných podmienkach¹.

PID riadenie bolo vytvorené v prostredí Arduino IDE pre prototypizačnú dosku Arduino DUE, kde vzorkovacia periódka pre zaznamenávanie dát bola nastavená na $T_s = 0.0015$ s. Ako som uviedol, pri vývoji a modelovaní sú uvažované 2 magnety. Moja osobná preferencia je 8 mm priemer (menší magnet) a s ním aj uvedené porovnanie realizujem, nakoľko trubica v ktorej magnet levituje má vnútorný priemer len 10 mm. Väčší magnet s priemerom 9 mm by sa mohol aj pri nekvalitnom riadení stabilizovať dotykom o stenu a tým by sme dostali skreslené výsledky. Ladenie PID regulátora je vzhľadom na Rov. (6.40) nasledovné:

$$K_P = 3.5,$$

$$T_i = 0.2,$$

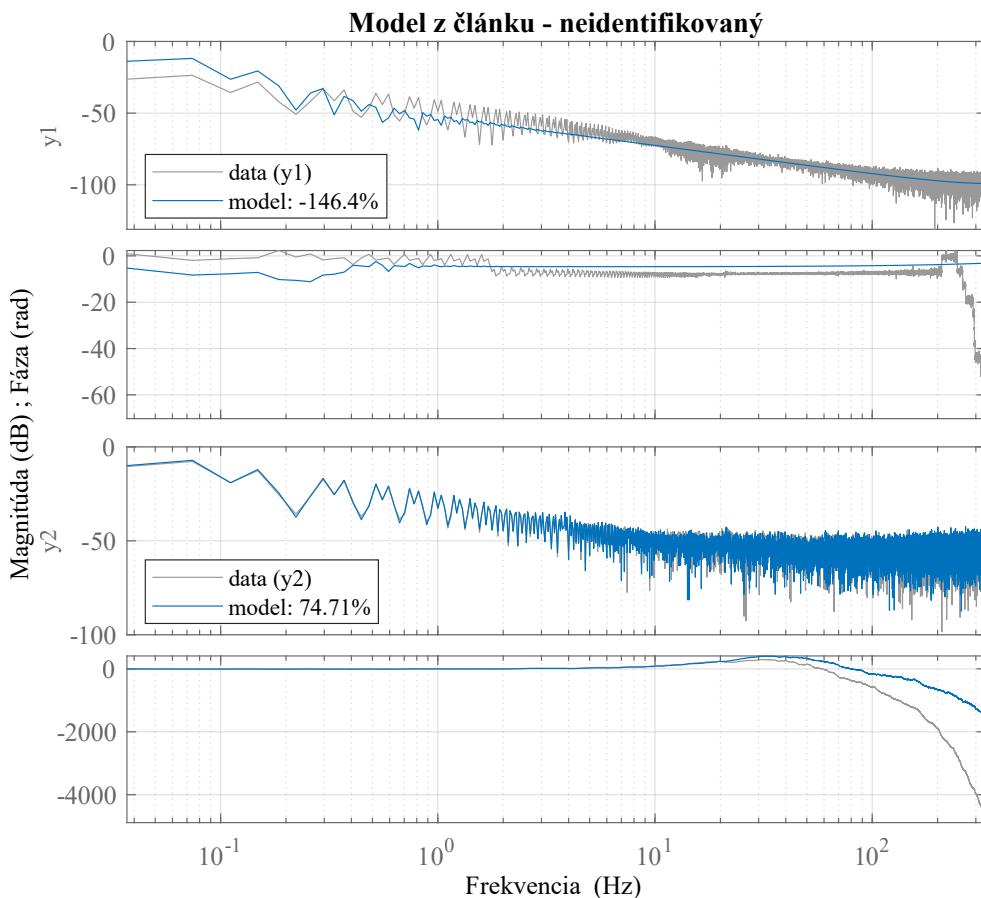
$$T_d = 0.025.$$

Ako prvé som porovnal samotný koncept modelov, nakoľko dobre vystihujú namerané dáta, bez procesu odhadu parametrov, len na základe údajov, ktoré som o systéme zozbieral, prípadne

¹Prvý model, ktorý bol uvedený už v článku, bol identifikovaný na základe experimentu s magnetom s priemerom 9 mm, ale môj model uvažuje 8 mm magnet. Síce by to nemalo mať výraznejší vplyv na výsledky, ale objektívne porovnanie vyžaduje totožné podmienky pre oba modeli.

vypočítať². Toto porovnanie som urobil v prostredí MATLAB, pomocou funkcie `compare()`, ktorá porovnáva nakoľko sa percentuálne zhodujú zaznamenané a simulované výstupy systému pri totožných vstupných dátach. Na Obr. 7.1 a 7.2 je možné vidieť výsledky tohto porovnania. Z neho je zrejmé, že navrhnutý model z Kap. 5 má omnoho vyššiu zhodu s nameranými dátami než má model z článku.

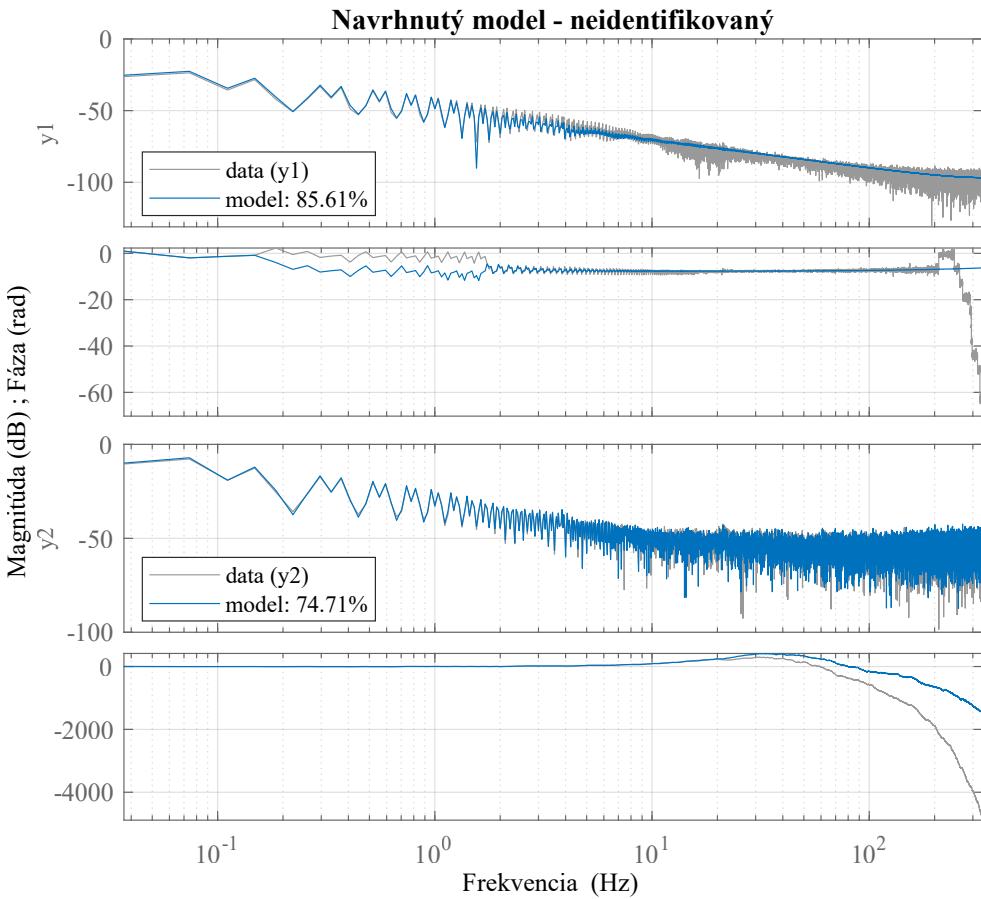
Avšak model, ktorý sa používa pri levitácii je po procese identifikácie. Preto je pre nás podstatnejšie, či navrhnutý model je lepší ako model z článku aj v prípade, že sme už vykonali odhad parametrov. Výsledky tohto porovnania sú na Obr. 7.3 a 7.4. Z porovnania je zrejmé, že oba modely veľmi dobre opisujú chovanie sa elektrického obvodu, avšak opäť, pri opise mechanického pohybu je model z Kap. 5 presnejší.



Obr. 7.1: Porovnanie modelu z článku s dátami z PID regulácie bez vykonania odhadu parametrov.

Okrem tohto porovnania som vykonal ešte jeden experiment. Ten spočíval v tom, že k obom systémom v ich diskrétnej forme som pridal PID riadenie, ktoré je použité pri riadení reálneho

²Tu môže vzniknúť určitá číselná rozdielnosť medzi hodnotami konštánt, ktoré sa nachádzajú na stránke projektu a sú použité v článku a medzi tými, ktoré uvediem tu, nakoľko naša znalosť o systéme od uvedenia článku pokročila. Napr. konštantu K pre model v článku pôvodne uvažuje relatívnu permeabilitu oceli $\mu_r = 150$. Avšak neskôr sa ukázalo, že jadro elektromagnetu je z materiálu podobnému čistému železu s relatívou permeabilitou okolo $\mu_r = 5000$

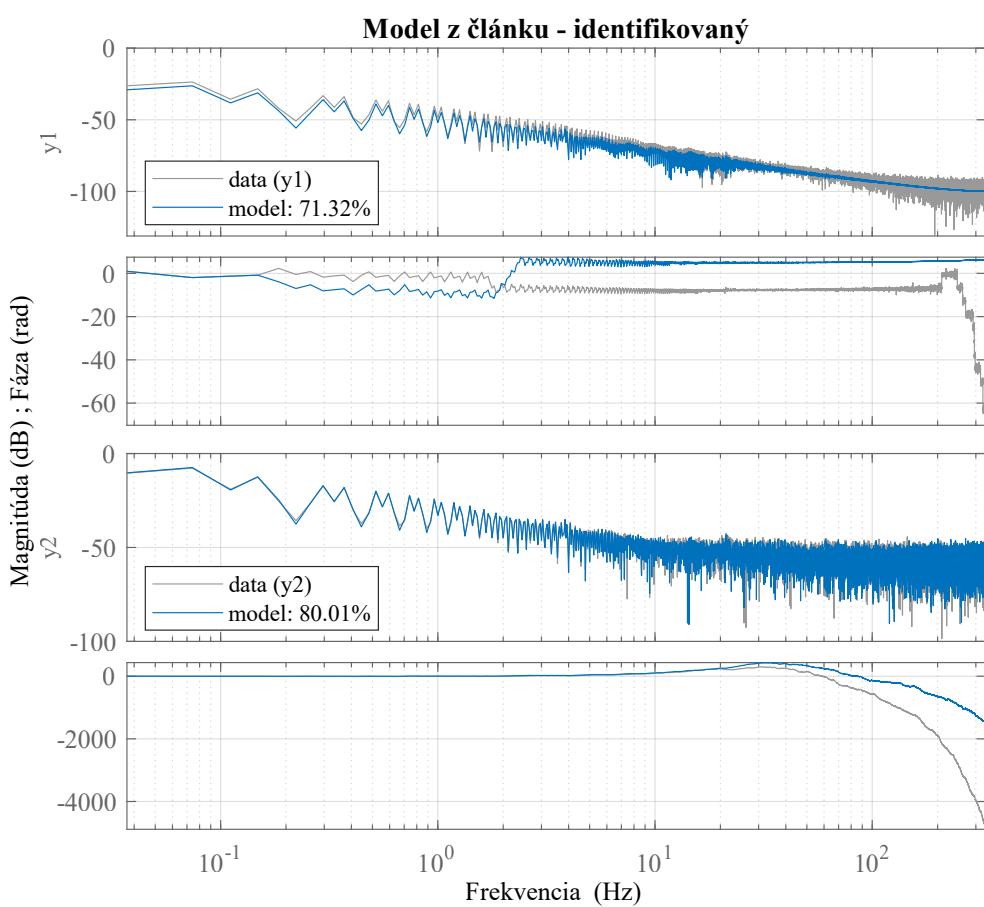


Obr. 7.2: Porovnanie navrhnutého modelu z Kap. 5 s dátami z PID regulácie bez vykonania odhadu parametrov.

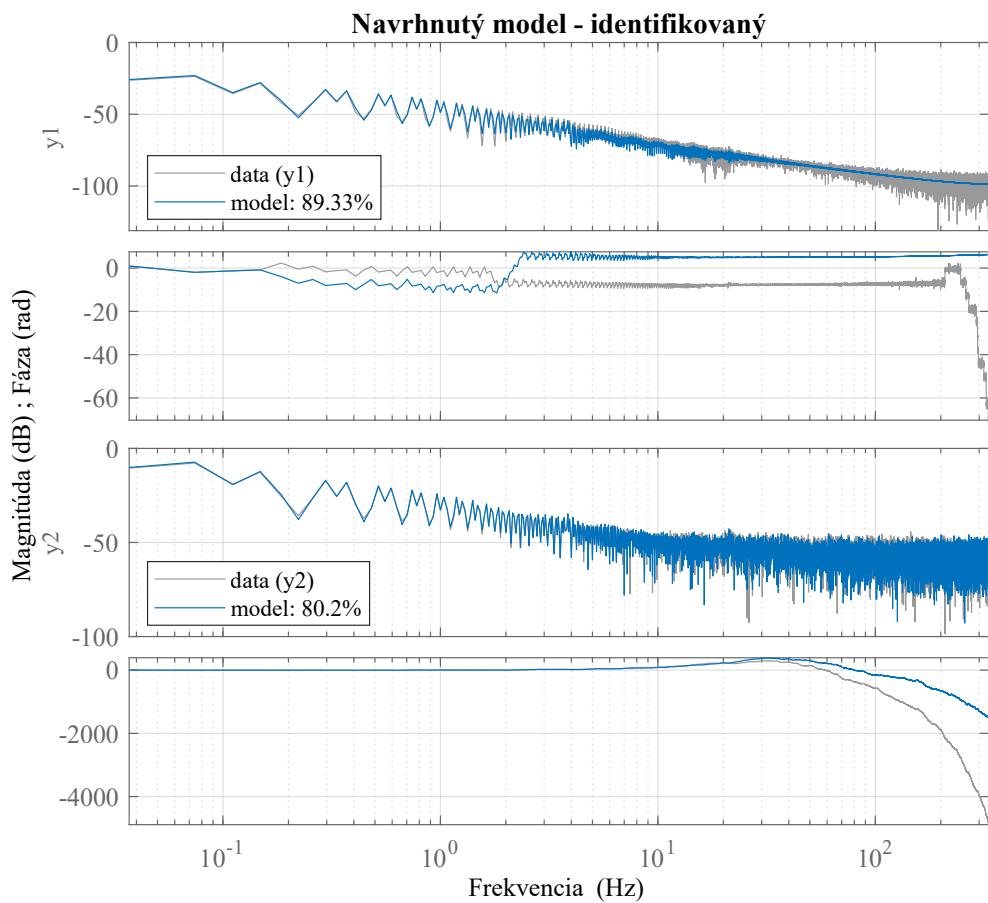
systému MagnetoShield v Kap. 6.4. Toto porovnanie má za cieľ ukázať, nie len či sú modely použiteľné pri návrhu riadenia, ale aj to, či sú systému natoľko podobné, že na ich základe môžeme riadenie aj otestovať. Za týmto účelom som použil riadiaci program zobrazený na Obr. 7.5. Postupne som do bloku reprezentujúceho diskrétny stavový priestor nahral najprv model z článku a následne navrhnutý model z Kap. 5. Výsledné priebehy sú zobrazené na Obr. 7.6 a Obr. 7.7.

V každom prípade, ako som už spomíнал, sú oba modely použiteľné. Ale ak vezmeme do úvahy, že za štruktúrou modelu a identifikovanými parametrami hľadáme aj určitú (zmysluplnú) fyzikálnu reprezentáciu, zákonitosť opísané modelom v článku nie sú presné³ a určite by som, na základe uvedených výsledkov, odporúčal pri ďalšej práci zvážiť použitie novonavrhnutého modelu z Kap. 5. Z tohto dôvodu sa posnažím v ďalšej podkapitole analyzovať, či je možné tento model využiť aj v jeho nelinearizovanej forme reprezentovanej Rov. (5.28).

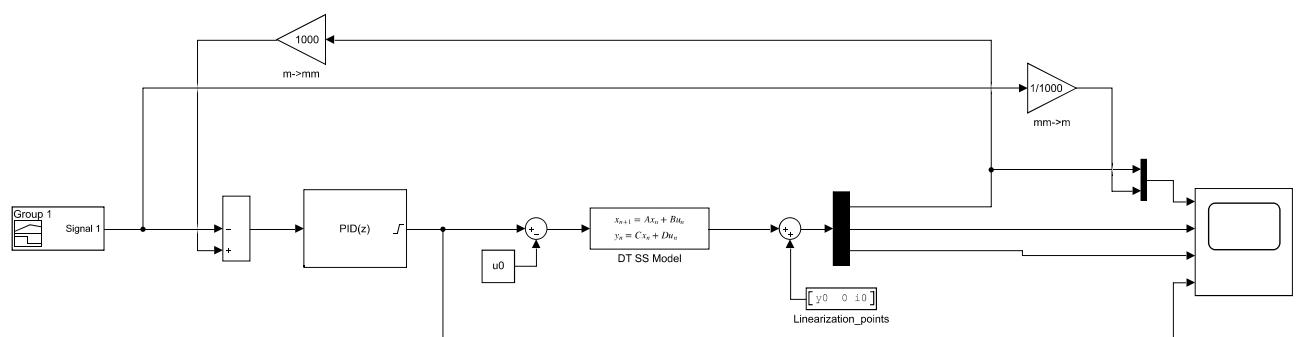
³Model z článku nie je presný pri opise systému MagnetoShield. Fyzikálna podstata, ktorú ale reprezentuje môže byť presná pri použití iného levitátora, prípadne po nahradení permanentného magnetu feromagnetickým telesom, nakoľko práve toto je jeden zo základných rozdielov, ktorým sa MagnetoShield líši od iných elektromagnetických levitátorov.



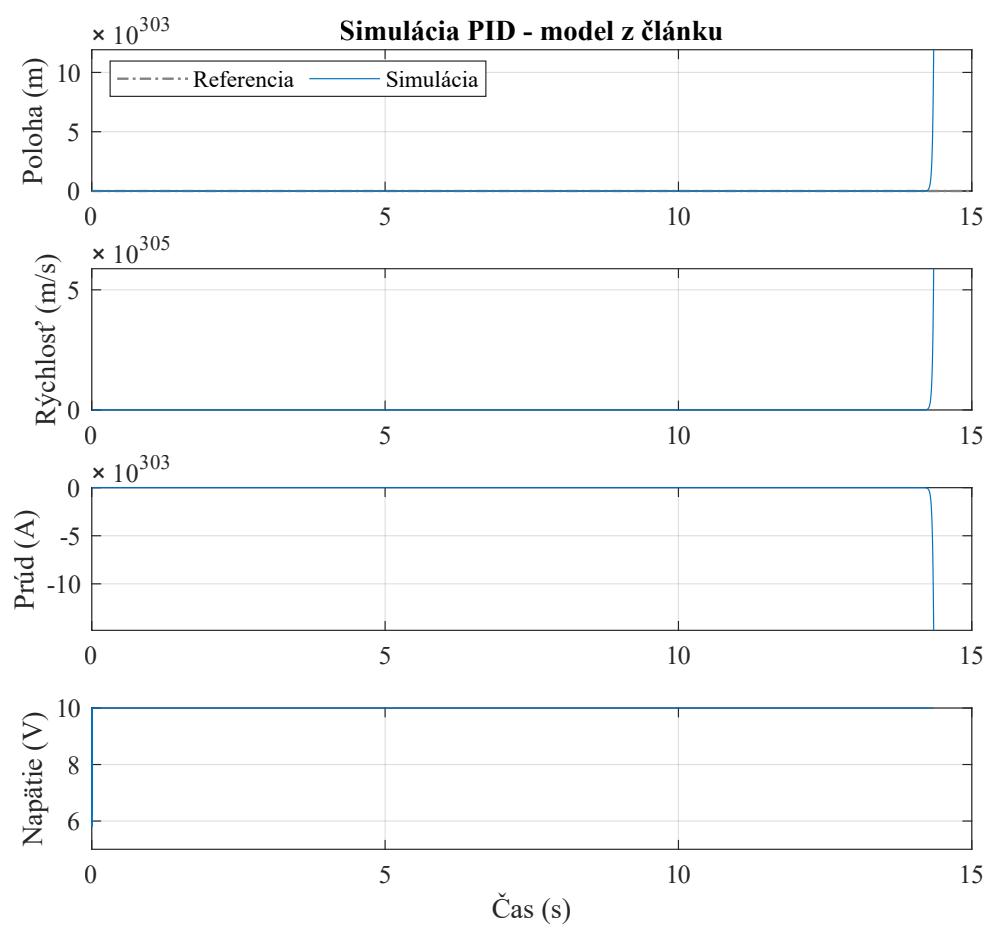
Obr. 7.3: Porovnanie modelu z článku s dátami z PID regulácie po vykonaní odhadu parametrov.



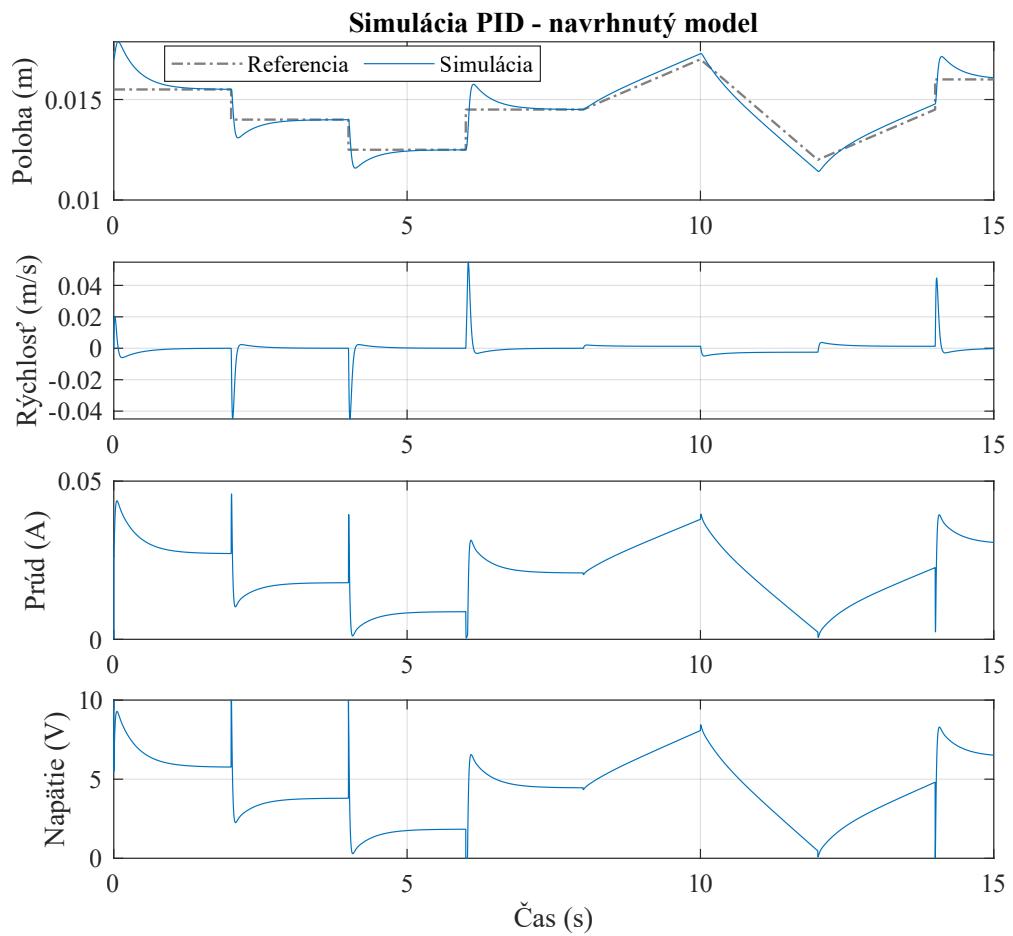
Obr. 7.4: Porovnanie navrhnutého modelu z Kap. 5 s dátami z PID regulácie po vykonaní odhadu parametrov.



Obr. 7.5: Simulované PID riadenie s použitím lineárneho modelu systému.



Obr. 7.6: Priebeh simulácie PID regulácie modelu z článku.



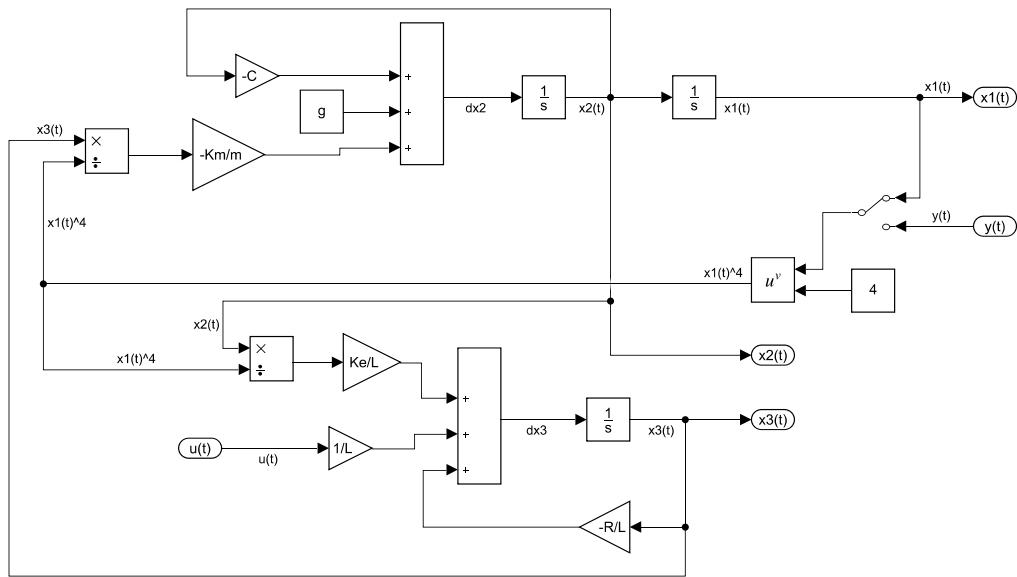
Obr. 7.7: Priebeh simulácie PID regulácie navrhnutého modelu z Kap. 5.

7.2 Nelineárny model

V predošej podkapitole som ukázal, že navrhnutý model je v lineárnej podobe presný a po procese identifikácie veľmi dobre charakterizujúci systém. V tejto časti sa posnažím o približnej analýze jeho využiteľnosti v nelineárnej forme.

Za účelom porovania nelineárneho modelu s fyzickým systémom, som v rámci prostredia Simulink vytvoril schématickú reprezentáciu rovnice Rov. (5.28), ktorá je na Obr. 7.8⁴. Taktiež som analogicky vytvoril túto reprezentáciu v diskrétnej podobe pre dané vzorkovanie T_s zobrazenej na Obr. 7.9, ktorá bola odvodená z Rov. (5.28) doprednou Eulerovou metódou pričom sme získali nasledovné diferenčné rovnice:

$$\begin{aligned} x_1[k+1] &= T_s x_2[k] + x_1[k], \\ x_2[k+1] &= T_s g - T_s K_m \frac{x_3[k]}{mx_1[k]^4} + (1 - T_s c)x_2[k], \\ x_3[k+1] &= \left(1 - \frac{T_s R}{L}\right)x_3[k] + \frac{T_s K_e x_2[k]}{Lx_1[k]^4} + \frac{T_s}{L} u[k]. \end{aligned} \quad (7.1)$$

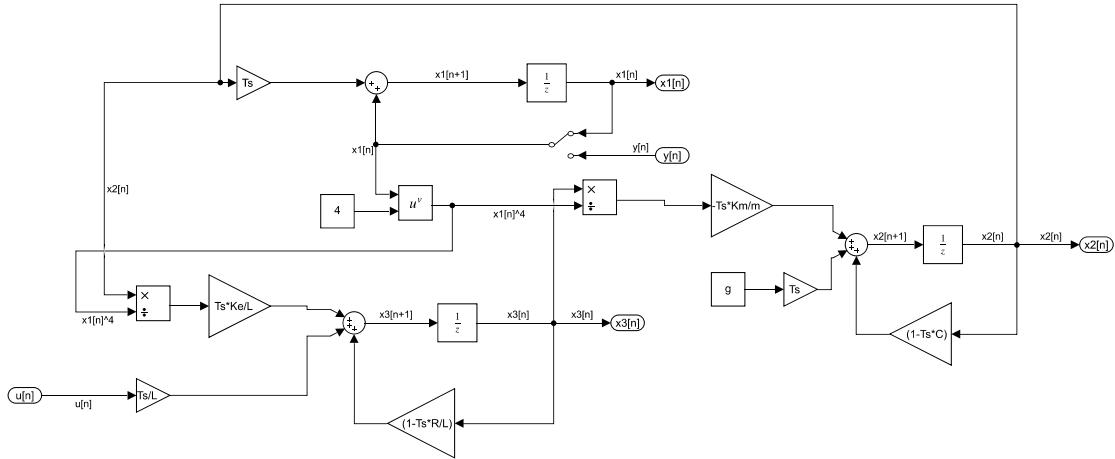


Obr. 7.8: Nelineárny model systému v prostredí Simulink pre spojity čas CT.

Princíp experimentu potvrdzujúci použiteľnosť a presnosť nelineárneho modelu je totožný s druhým experimentom v minulej podkapitole a teda, na uvedený vymodelovaný systém aplikujem PID riadenie s totožným ladením, ako je tomu pri reálnom systéme. Následne výsledné dátá porovnám so skutočným experimentom. Nakoľko je PID na reálnom systéme diskrétnie, budem využívať DT reprezentáciu modelu uvedenú v Rov. (7.1) a zobrazenú schématicky na Obr. 7.9.

Pred implementáciou riadenia na nelineárny systém je nutné zvážiť určité možné rozdielnosti v hodnotách identifikovaných konštánt. Hodnoty pre linearizovaný model v tabuľke 5.3 boli odhadované na základe lineárneho stavového systému reprezentovaného maticovou formou v Rov.

⁴ V schéme je vidieť možnosť posielania dát polohy z externého zdroja. Vďaka tomu možno z túto reprezentáciu použiť aj ako stavový pozorovateľ.



Obr. 7.9: Nelineárny model systému v prostredí Simulink pre diskrétny čas DT.

(5.40). Ako som spomíнал, odhad parametrov nehľadá hodnoty konkrétneho parametra, ale koeficienty matice a z nich je následne možné daný parameter dopočítať. Z toho a z Rov. (5.40) vyplýva, že hodnota parametra je nie závislá len od vlastností systému, ale aj od voľby linearizačných bodov x_{01} , x_{02} a x_{03} (tie boli zvolené pevne a nepodliehali procesu odhadu). Takže použitie totožných hodnôt pre konštanty a parametre systému (napr. K_e , K_m) pri nelineárnom modeli nemusia vystihovať chovanie sa systému a môžu byť nesprávne.

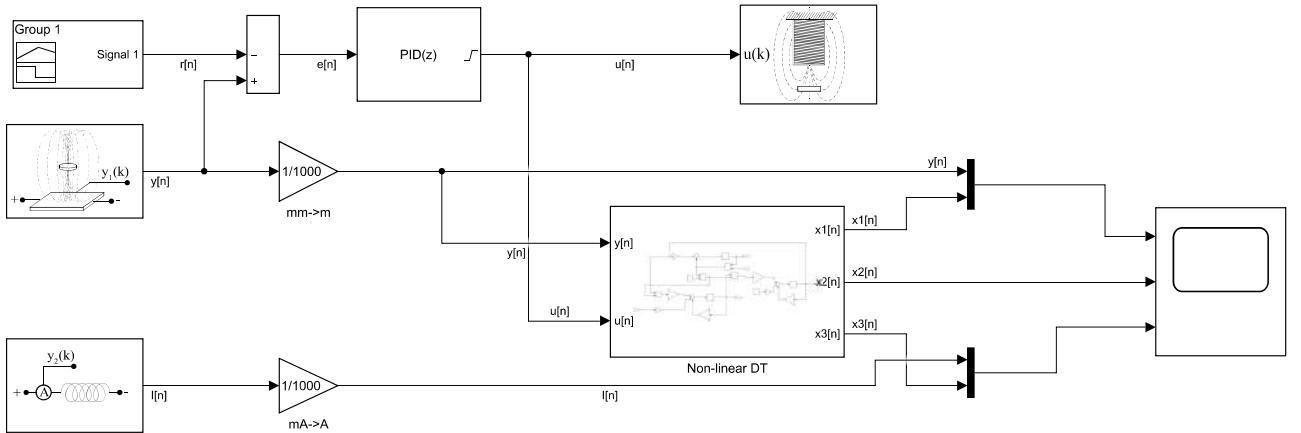
Z tohto dôvodu by bolo ideálne, celý proces identifikácie pre nelineárny model zopakovať. Avšak nakoľko to je komplikovaný proces, rozhodol som sa, že identifikované parametre overím a prípadne doladím ich hodnotu iným spôsobom. Za týmto účelom som opäť použil PID riadenie s istým typom "nelineárneho pozorovateľa".

Riadenie systému prebieha úplne totožne, ako to bolo opísané v Kap. 6.4 s tým, že vstup je taktiež posielaný do DT nelineárneho modelu existujúceho paralelne s obvodom riadenia (Obr. 7.10). Ak posielame ale do modelu len vstupné hodnoty počítané na základe reálneho systému, chovanie modelu by sa dalo opísať ako "open-loop" riadenie. V takomto prípade, ak by parametre nelineárneho modelu boli nesprávne, viedlo by to k nestabilite odhadovaných hodnôt týmto pozorovateľom. Z tohto dôvodu som neposielal do systému len vstupné napätie, ale aj reálne merania polohy (na Obr. 7.9 je prepínač v spätej väzbe pre výpočet stavu x_1 prepnutý na vstup $y[k]$). To zabezpečuje, že odhad nebude divergovať a súčasne budeme schopní vidieť nesprávnu voľbu parametrov na stavoch rýchlosť a prúdu (prúd v elektromagnete vieme merať, tým pádom vieme povedať, či je jeho odhad zlý a o hodnotách rýchlosť si vieme urobiť predstavu z merania v minulých kapitolách napr. z LQ riadenia s použitím Kalmanovho filtra zobrazenom na Obr. 6.19).

S použitím uvedeného experimentu som upravil hodnotu konštanty K_m spôsobom postupnej zmeny parametra a opakovaného spúšťania experimentu na hodnotu

$$K_m = 1.8752 * 10^{-8}.$$

Hodnoty ostatných konštant som nechal s tými výsledkami, ktoré boli určené v procese identifikácie linearizovaného systému. Výsledné porovnanie nameraných hodnôt jednotlivých stavov s odhadnutými hodnotami stavov je vidieť na Obr. 7.11. Ako vidíme, hodnota polohy je podľa očakávaní veľmi podobná reálnym meraniam, nakoľko tie boli v spätej väzbe vnášané do neline-



Obr. 7.10: PID riadenie s “nelineárnym pozorovateľom”.

árneho modelu. Hodnota rýchlosťi, z ktorej máme len odhad⁵, má pomerne zmysluplné hodnoty, nakoľko sa pohybuje okolo nuly a nenadobúda obrovské rozmery. Hodnota prúdu je veľmi podobná meraniam, charakter priebehu je totožný a rozdiel je skutočne minimálny.

Nakoľko som menil jediný parameter a to konštantu K_m , ktorá by podľa uvedenej teórie v Kap. 5 mala byť totožná s konštantou K_e , je možné uvažovať, či naozaj používam správne hodnoty jednotlivých parametrov (nakoľko K_m a K_e sa líšia) a či nie je možné nájsť model s konštantami opisujúcimi skutočnosť viero hodnejšie.

Pravdou je, ako som už naznačil, že určite áno. Jednoznačne by bola ideálna opäťovná analýza modelu, významu jednotlivých parametrov a ich hodnôt v kontexte fyzikálnych zákonitostí a informácií, ktoré máme o systéme. Taktiež by bolo v budúcnosti dobré zvážiť opäťovný dopad konštanty tlmenia c , ktorá bola v rámci linearizovaného modelu určená ako nulová, či nemá významnejší vplyv v rámci nelineárneho systému.

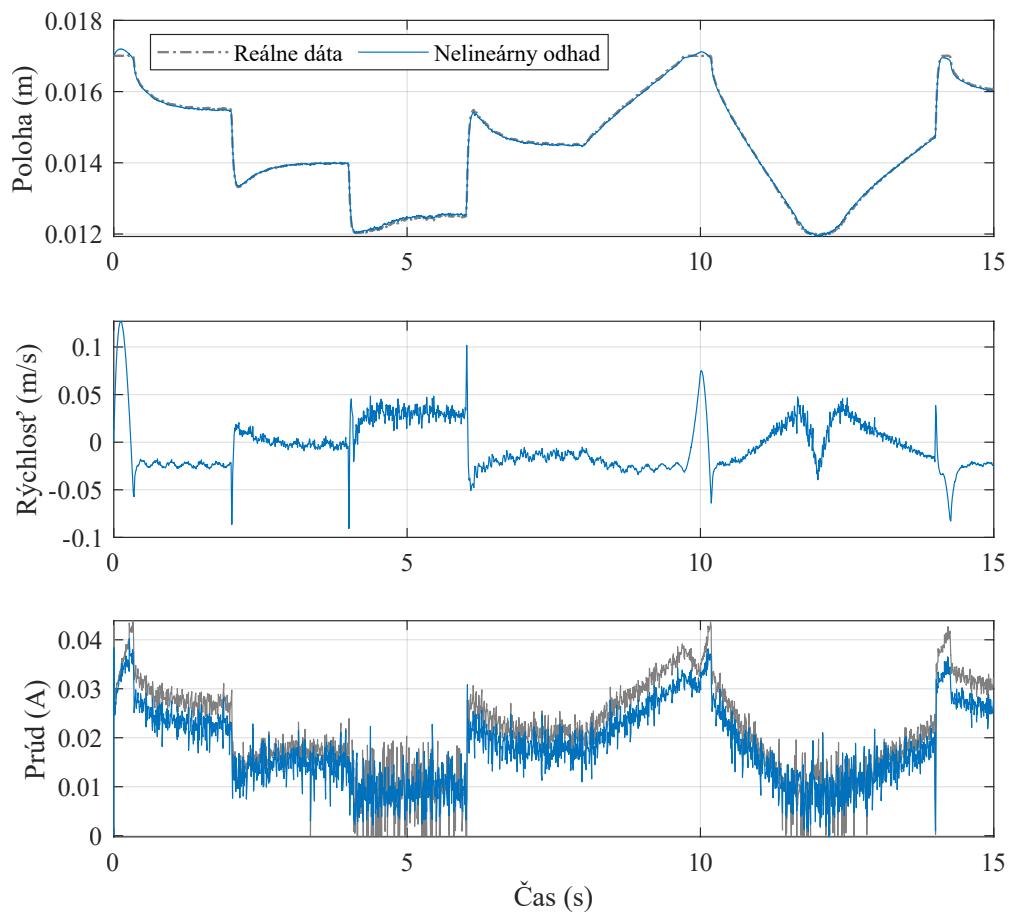
Napriek tomu, dosiahnuté výsledky nelineárneho modelu z Obr. 7.11 naznačujú, že navrhnutý model by mohol byť správny.

Nakoniec som realizoval konečný experiment, kedy som k modelu nelineárneho systému pridal PID riadenie s ladením, použitým pri skutočnom riadení MagnetoShieldu. Schéma tejto simulácie je na Obr. 7.12 a výsledný priebeh veličín je na Obr. 7.13.

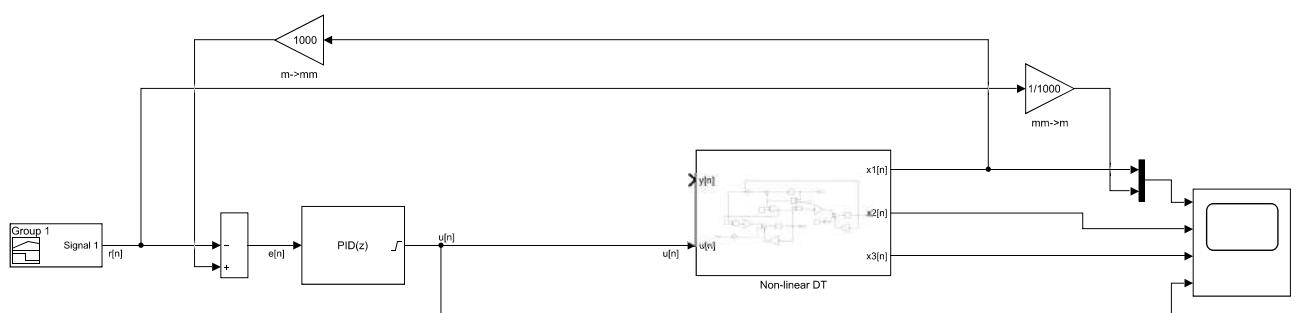
Ako vidíme, nie len že s totožným ladením sme schopní systém uriadiť, ale pri porovnaní s priebehom polohy pri skutočnom experimente zobrazenom na Obr. 7.11 je vidieť, že aj samotný charakter a chovanie sa priebehu je prakticky rovnaké. Povzbudivými sú aj porovnania rýchlosťi a prúdu s Obr. 7.11, ktoré sú taktiež veľmi podobné. Podobný experiment by bolo možné uskutočniť aj s ďalšími typmi riadení, nie len PID.

Výsledkom tejto podkapitoly je teda, že model ktorý som v rámci identifikácie navrhol je použiteľný pre opis systému aj v jeho nelineárnej forme a aj keď by bolo dobré vykonať ešte analýzu a spresnenie hodnôt parametrov, už improvizované ladenie, ktoré bolo uvedené, umožňuje využívanie tohto modelu pre účely testovania, riadenia alebo odhadu stavov systému.

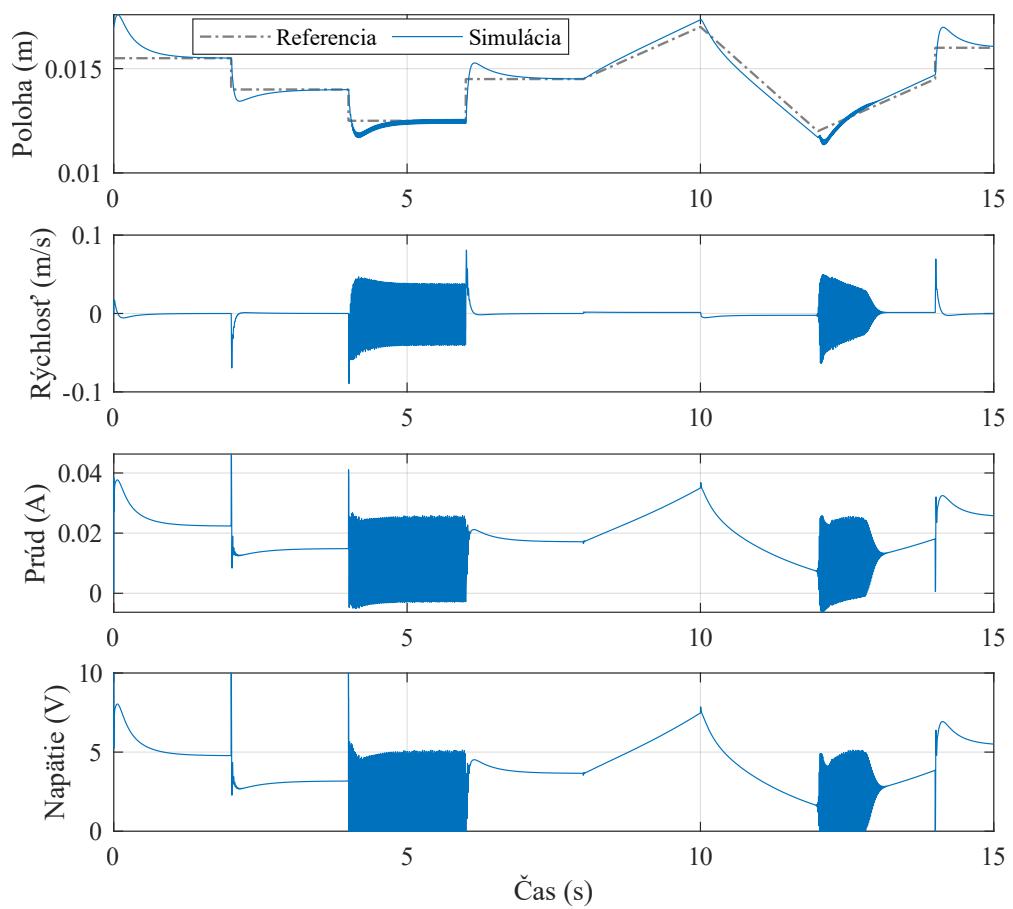
⁵Priame meranie rýchlosťi na MagnetoShield nie je možné.



Obr. 7.11: Priebehy nameraných stavov a ich odhadnuté hodnoty pomocou nelineárneho pozorovateľa.



Obr. 7.12: Simulované PID riadenie nelineárneho modelu systému.



Obr. 7.13: Priebeh simulácie PID riadenia nelineárneho modelu MagnetoShieldu s použitím ladenia zo skutočného systému.

8 Záver

Vývoju zariadenia MagnetoShield som sa venoval, nie len v predošlých kapitolách, ale aj vo svojej bakalárskej práci. Výsledkom tohto úsilia, spolu s prácou a časom ktorý tomu venovali prof. Ing. Gergely Takács a Ing. Erik Mikuláš, je hotové a funkčné zariadenie, ktoré pomocou elektromagnetu a spätnoväzobného riadenia dokáže vyzdvihnuť a udržať permanentný magnet vo vzduchu. Okrem samotného zariadenia boli vytvorené na jeho ovládanie viaceré aplikačné rozhrania, z ktorých najväčší význam majú v rámci vzdelávania študentov Arduino IDE a Simulink. Taktiež boli vytvorené mnohé ukážkové príklady, reprezentujúce najrôznejšie typy riadenia na jednotlivých rozhraniach a príklady analyzujúce systém z pohľadu jeho fyzikálnych vlastností. Všetky poznatky a príklady, ktoré boli v súvislosti so systémom zistené, ak aj neboli spomenuté v tejto práci, sú prístupné na stránke projektu [19].

Z môjho osobného hľadiska je MagnetoShiled hotový a má dostatočne robustné zázemie informácií na to, aby bol aj aktívne využívaný. Napriek tomu by som rád v závere tejto práce spomenul niekoľko "nápadov", ktoré by mohli stať za zváženie v prípade ďalšieho rozvoja zariadenia.

- Ako bolo viac krát počas práce uvádzané, jedná sa o dynamické zariadenie a vzorkovacie periody sú využívané takmer v maximálnych frekvenciach. Kompatibilita shieldu s výkonnejšou výpočtovou technikou by výrazne zjednodušila návrh riadiacich programov, nakoľko by nebolo nutné brať do úvahy obmedzenia prototypizačných dosiek Arduino a vzniklo by viac priestoru pre experimentovanie s vlastnosťami systému.
- Ďalšou možnosťou, ktorá by umožnila vytváranie veľmi komplexných riadiacich algoritmov je riadenie, kde výpočtový program beží externe, mimo Arduina. Následne medzi Arduinom a externým zariadením prebieha jedine komunikácia obsahujúca záznam o namenaných hodnotách veličín a pokyny pre akčný člen, aké napäťie má byť generované na elektromagnete. Nakoľko sú vývojové prostredia MATLAB a Simulink pre návrh a analýzu riadenia výborne vybavené, odporúčam tento koncept preskúmať pre vývojové prostredie Simulink v prípade, že riadenie má vysoké nároky na vzorkovanie ($T_s \leq 1 \text{ ms}$). Podobný koncept som navrhol aj v závere Obr. 4.2.1.
- Preskúmanie pamäťovej a výpočtovej optimálnosti funkcií použitých v knižničach AutomationShield pre rozhranie Arduino IDE. Pre niektoré riadiace programy je možné dosiahnuť vyššie vzorkovacie frekvencie v prostredí Simulink, kde je zdrojový kód automaticky generovaný. Na základe toho možno predpokladať, že knižnice nie sú pamäťovo a výpočtovo optimalizované. Toto prípadné zlepšenie by rozšírilo možnosti aplikácie zložitejších riadiacich algoritmov na jednoduchšie výpočtové zariadenia. Či by ale došlo k reálnemu zlepšeniu, nie je možné teraz s istotou tvrdiť, nakoľko samotné Arduino IDE obashuje sekvenčiu prednastavených operácií rozširujúcich zdrojový kód, na ktoré bežný používateľ nemá dosah.
- Značná časť mojej práce bola venovaná návrhu modelu a jeho následnému použitiu. Samotný model sa preukázal byť po linearizácii veľmi presný. Preto som sa snažil tento

model porovnať s prevzatým modelom, ktorý bol pri práci s MagnetoShieldom využívaný primárne. Z realizovaných testov vyšiel mnou navrhnutý model jednoznačne lepšie. Avšak napriek tomu, by som odporúčal otestovať presnosť linearizovaného modelu aj pri zmene-nom hardvéry, či jeho presnosť naložaj opisuje takýto systém všeobecne. Určite je nutné, aby základný koncept ostal zachovaný a teda elektromagnet vytvárajúci nie príliš silné magnetické pole príťahuje extrémne silný neodymový magnet s tým, že ich vzájomná vzdialenosť nie je príliš malá. Navrhoval by som otestovať využitie elektromagnetu, pri ktorom sú jeho fyzikálne a konštrukčné vlastnosti presne uvedené a nie je nutné ich experimentálne určovať. Tým myslím predovšetkým počet závitov N použitých na elektromagnete a materiál z ktorého je vyrobené jadro elektromagnetu, nakoľko to definuje remanenciu B_r , ktorá taktiež zohráva zásadnú úlohu pri určení magnetickej konštanty systému K . Taktiež prínosným experimentom by bolo otestovanie systému, pri ktorom sú použité neodymové magnety s rôznou hmotnosťou a či presné definovanie hmotnosti zlepšuje presnosť modelu.

- Veľmi prínosným zistením tejto práce je, že mnou navrhnutý nelineárny model sa pri riadení chová podobne reálnemu systému (otestované zatial len pre PID). Z tohto dôvodu by som zvážil, pri zobrať do úvahy aj vyššie opísané odporúčané experimenty, navrhnut a realizovať nelineárnu identifikáciu parametrov. Takýto identifikovaný model by otváral nové možnosti experimentovaniu so systémom v rámci nelineárnych spôsobov riadenia.
- Poslednou navrhnutou zmenou, ktorú by som zvážil je už spomenutá zmena magnetu. Originálne bol uvažovaný permanentný neodymový magnet o priemere 8 mm. Táto veľkosť bola opodstatnená, nakoľko vnútorný priemer trubice je 10 mm a tým pádom mal magnet dostatočný priestor okolo seba, aby nenarážal do jej stien, čo by napomáhalo stabilizácii magnetu aj v prípade nesprávne navrhnutého riadenia¹. Pri experimentoch a vytváraní prediktívneho riadenia bol ale tento magnet vymenený za magnet s 9 mm priemerom. Dôvodom bolo, že bol ľažší, čo malo zmenšovať efekt prípadnej oscilácie okolo ustálenej hodnoty. Avšak priestor okolo magnetu je menší, čo môže viesť ku chybným meraniam. Naviac, ak bude využívaný správny model systému, váha magnetu by nemala mať vplyv na priebeh levitácie, nakoľko model berie túto skutočnosť do úvahy. Každopádne väčší magnet je tiež možnosťou, ale je nutné dávať obzvlášť pozor, či je MagnetoShield postavený rovno a levitujúci magnet sa netrie o stenu trubice.

Iniciatíva, ktorej je MagnetoShield súčasťou, je od študentov pre študentov a účelom týchto zariadení je, aby boli využívané ako edukačné pomôcky. Niečo, na čom študent odskúša a upresní svoje vnímanie nadobudnutých poznatkov. Niečo, čo pojmu abstrakcia a reprezentácia, dokáže dať fyzickú podobu. Niečo, čo prepojí naučené s reálnym a odhalí, že teória nie je učením sa pre učenie, ale je rovnako skutočná ako to, čo popisuje. S týmto cieľom bolo zariadenie MagnetoShield navrhnuté a skonštruované. Preto pevne verím, že bude aj týmto spôsobom využívané a každý nový poznatok o tomto, či inom zariadení neupadne do nezáujmu alebo nepozornosti. Lebo iba spoločné hľadanie objektívnej pravdy, skúmanie, ako veci okolo nás fungujú a voľné zdielanie týchto poznatkov, vedia hýbať spoločnosťou vpred.

¹Ideálnym by bolo permanentné odstránenie trubice, tá má ale za úlohu používateľovi pomáhať, aby magnet nestratil v prípade, že riadenie nie je správne navrhnuté. V prípade že je riadenie otestované a stabilné, je možné odstrániť aj trubicu.

Literatúra

- [1] J. Ackermann. Der entwurf linearer regelungssysteme im zustandsraum. *at - Automatisierungstechnik*, pages 297–300, 1972.
- [2] B.D. Cullity and C.D. Graham. *Introduction to Magnetic Materials*. Wiley, 2009.
- [3] M. Sami Fadali and Antonio Visioli. Chapter 10 - optimal control. In M. Sami Fadali Fadali and Antonio Visioli, editors, *Digital Control Engineering (Second Edition)*, pages 399–438. Academic Press, Boston, second edition edition, 2013.
- [4] M. Sami Fadali and Antonio Visioli. Chapter 8 - properties of state-space models. In M. Sami Fadali and Antonio Visioli, editors, *Digital Control Engineering (Second Edition)*, pages 293–349. Academic Press, Boston, second edition edition, 2013.
- [5] FIRST4MAGNETS. Neodymium magnet grades. [Online] <https://www.first4magnets.com/tech-centre-i61/information-and-articles-i70/neodymium-magnet-information-i82/grades-of-neodymium-magnets-i92>. [24. Mar. 2021].
- [6] Emilio Frazzoli and Munther Dahleh. 6.241J Dynamic Systems and Control. Massachusetts Institute of Technology: MIT OpenCourseWare, Spring 2011. <https://ocw.mit.edu> License: Creative Commons BY-NC-SA.
- [7] Dennis Freeman. 6.003 Signals and Systems. Massachusetts Institute of Technology: MIT OpenCourseWare, Fall 2011. <https://ocw.mit.edu> License: Creative Commons BY-NC-SA.
- [8] Carlo Gonano, Riccardo Zich, and Marco Mussetta. Definition for polarization p and magnetization m fully consistent with maxwell's equations. *Progress In Electromagnetics Research B*, 64:83–101, 11 2015.
- [9] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*, pages 502–510, Zürich, Switzerland, July 17–19 2013. <http://control.ee.ethz.ch/~mpt>.
- [10] F.S. Hillier and G.J. Lieberman. *Introduction to Operations Research*. McGraw-Hill International Editions. McGraw-Hill, 2001.
- [11] W. G. Hurley and W. H. Wolfle. Electromagnetic design of a magnetic suspension system. *IEEE Transactions on Education*, 40(2):124–130, 1997.
- [12] Hyung-Woo Lee, Ki-Chan Kim, and Ju Lee. Review of maglev train technologies. *IEEE Transactions on Magnetics*, 42(7):1917–1925, 2006.
- [13] Michel Levis. Quanser Inc., Sep 2018. <https://www.quanser.com/blog/nonlinear-system-for-intermediate-control/>.

- [14] Lalo Magni, Davide Martino Raimondo, and Frank Allgöwer, editors. *A Survey on Explicit Model Predictive Control*, pages 345–369. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [15] J. Mihalík. Miniaturizácia experimentu magnetická levitácia. Slovenská technická univerzita v Bratislave, Máj 2018. Bakalárská práca.
- [16] D.S. Naidu. *Optimal Control Systems*. Electrical Engineering Series. Taylor & Francis, 2002.
- [17] James Roberge, Joel Dawson, and Kent Lundberg. 6.302 Feedback Systems. Massachusetts Institute of Technology: MIT OpenCourseWare, Spring 2007. <https://ocw.mit.edu> License: Creative Commons BY-NC-SA.
- [18] H. Semat and R. Katz. *Physics*. Rinehart, 1958.
- [19] G. Takács, P. Chmurčiak, R. Koplinger, T. Konkoly, G. Penzinger, M. Gulan, J. Kulhánek, L. Vadovič, M. Biro, E. Vargová, M. Vríčan, and J. Mihalík. AutomationShield. <https://github.com/gergelytakacs/AutomationShield>, 2021.
- [20] G. Takács, J. Mihalík, E. Mikuláš, and M. Gulan. Magnetoshield: Prototype of a low-cost magnetic levitation device for control education. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1516–1525, 2020.
- [21] Gergely Takács, Peter Chmurčiak, Martin Gulan, Erik Mikuláš, Jakub Kulhánek, Gábor Penzinger, Marcel Vdoleček, Miloš Podbielančík, Martin Lučan, Peter Šálka, and Dávid Šroba. Floatshield: An open source air levitation device for control engineering education. *IFAC-PapersOnLine*, 53(2):17288–17295, 2020. 21th IFAC World Congress.
- [22] Gergely Takács and Martin Gulan. *Základy prediktívneho Riadenia*. Spektrum STU, Bratislava, Slovakia, 1. edition, 2018. In Slovak language. (Fundamentals of Predictive Control).
- [23] Gergely Takács, Martin Gulan, Juraj Bavlna, Richard Köplinger, Michal Kováč, Erik Mikuláš, Sohaibullah Zarghoon, and Richard Salíni. Heatshield: a low-cost didactic device for control education simulating 3d printer heater blocks. In *2019 IEEE Global Engineering Education Conference (EDUCON)*, pages 374–383, 2019.
- [24] Gergely Takács, Tibor Konkoly, and Martin Gulan. Optoshield: A low-cost tool for control and mechatronics education. In *2019 12th Asian Control Conference (ASCC)*, pages 1001–1006, 2019.
- [25] A. Tirpák. *Elektromagnetizmus*. IRIS, Bratislava, Slovakia, 2014.
- [26] Engineering ToolBox. Permeability. [Online] Available at: https://www.engineeringtoolbox.com/permeability-d_1923.html, 2016. [Accessed 17. Mar. 2021].
- [27] Melda Ulusoy. Understanding Kalman Filters. MathWorks [online], 30. Jan 2017. <https://www.mathworks.com/videos/series/understanding-kalman-filters.html>.
- [28] Wen Yu and Xiaoou Li. A magnetic levitation system for advanced control education. *IFAC Proceedings Volumes*, 47(3):9032 – 9037, 2014. 19th IFAC World Congress.
- [29] K. Yung, P. Landecker, and D. Villani, Daniel. An analytic solution for the force between two magnetic dipoles. *Physical Separation in Science and Engineering*, 9:39–52, 1998.

- [30] Shengguo Zhang, Shuo Zhang, and Guoheng Zhang. Design of tracking control system by closed-loop pole placement. In *Proceedings of the 3rd International Conference on Mechatronics and Industrial Informatics*, pages 79–84. Atlantis Press, 2015/10.
- [31] P. Zometa, M. Kögel, and R. Findeisen. muAO-MPC: a free code generation tool for embedded real-time linear model predictive control. In *Proc. American Control Conference (ACC), 2013*, pages 5340–5345, Washington D.C., USA, 2013.